



Development of a Just-in-Time Railway Maintenance Support Tool

by
Henry Cobbold

Supervisor: Dr. Kemi Ademoye

Second Marker(s): Dr. Nicola Whitehead and Dr. Michael Dacey

Project submitted as part of the requirements for the award of
BSc(Hons) Applied Computing

April 2019

DECLARATION FORM AND FORM OF CONSENT

DECLARATION

I, **Henry T. G. Cobbold** declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed:

A handwritten signature in black ink, appearing to read 'H. Cobbold', is written over a faint, circular official stamp.

Date: 01/05/19

FORM OF CONSENT

I, **Henry T. G. Cobbold** hereby consent that my Project, submitted in candidature for the B.Sc. Applied Computing degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations

Signed:

A handwritten signature in black ink, appearing to read 'H. Cobbold', is written over a faint, circular official stamp.

Date: 01/05/19

ACKNOWLEDGEMENTS

I wish to firstly express my particular gratitude to *Dr. Kemi Ademoye* as project supervisor and the projects second-markers of *Dr. Nicola Whitehead* and *Dr. Michael Dacey* for their time, support and proof reading throughout. Your continual recommendations, praise and encouragement for high standards has been greatly appreciated and has pushed me to make this my best possible piece. This thanks also reaches out to all other lecturers with whom I have discussed elements of the project. Your input has also been much appreciated throughout.

Furthermore, I would like to extend my thanks to towards *Colin Page*, *Chris Lunn* and the rest of the GWR Staff in the Laira team for the time and effort they have invested in me during this project by signposting and directing me through their work. Without their help this project simply would not have been possible. To conclude, I am particularly grateful to *Chris Wade* for his assistance in providing a suitable channel through which to access industry professionals. Without his support in setting up me up with Colin and his team in Plymouth I would not have been able to manage this scenario.

ABSTRACT

The aim of the project is to develop an application for the rail industry that supports existing Just-in-Time e-procurement processes. Given the context of this study the practices and core themes of the paradigm will be explored and extended so as to provide a tool that can be used to manage maintenance and work scheduling practices in such a way that aligns with the core values of this deep-routed, and well documented method. To understand the viability, and potential of such a software tool the developer has partnered with a contemporary firm that whose work is based on the core issue of maintenance scheduling and component procurement activities.

To fulfil the goal of developing a useful and desirable software product the author has based the development of the system around the complex database system from which the tight operational schedule can be followed. To provide a user-friendly interface which supports the needs of the business the developer must consider several usability frameworks and concepts; ensuring that they are applied throughout the development of a front-end interface. Created with a lightweight template builder several rounds of developer and user-level testing will be performed so as to understand the suitability of the front-end tool and the skills of the developer in designing a front-end for a highly specific user group.

Following the development of an initial prototype, the product can be evaluated against the key issues of usability and Just-in-Time to understand how usable the product can be considered. By understanding its relative success, the author can stipulate as to why, and importantly how, the product can be extended in future development iterations

TABLE OF CONTENTS

DECLARATION FORM AND FORM OF CONSENT.....	I
ACKNOWLEDGEMENTS.....	II
ABSTRACT.....	III
TABLE OF CONTENTS.....	IV
TABLE OF FIGURES.....	VII
TABLE OF TABLES.....	IX
1. INTRODUCTION	1
1.1 CASE STUDY PROCESSES	1
1.1.1 Introduction.....	1
1.1.2 Existing Maintenance Processes.....	1
1.1.3 Physical Work Environment.....	4
1.1.4 Existing Ordering and Supply Chain Processes	5
1.1.5 Case Study Conclusions.....	5
1.2 PROJECT AIM.....	5
1.3 PROJECT OBJECTIVES	6
1.4 REPORT STRUCTURE	6
2. RESEARCH AND REVIEW OF LITERATURE.....	8
2.1 INTRODUCTION	VII
2.2 CONTEXT OF LITERATURE REVIEW	8
2.3 THE 'JUST-IN-TIME' CONCEPT.....	8
2.3.1 The History of 'Just-in-Time'	9
2.3.2 Concepts of 'Just-in-Time'	10
2.3.3 The Key is Kanban	13
2.3.4 Influence and Uptake	17
2.3.5 'Just-in-Time' Supply Chain Management	18
2.4 DESIGNING MOBILE APPLICATIONS.....	20
2.4.1 Uptake and Value of Mobile Devices.....	20
2.4.2 Usability of Mobile Applications	20
i. Defining Usability.....	20
ii. Usability in User Experience.....	21
3. RESEARCH AND DEVELOPMENT METHODOLOGIES	29
3.1 RESEARCH METHODOLOGY	29
3.1.1 Research Philosophy.....	29
3.1.2 Research Approach.....	30
3.1.3 Research Strategy.....	30
3.2 DATA COLLECTION.....	31
3.2.1 Collection Methods	31
3.2.2 Requirements Elicitation Processes.....	32
3.2.3 Data Governance	33
3.3 PROJECT DELIVERABLES	34
3.4 DEVELOPMENT METHODOLOGY	34
3.4.1 Agile Software Development.....	34
3.4.2 Feature-Driven Development.....	35
i. Concepts of Feature-Driven Development.....	35
ii. Project Scenario and Adapting FDD	36
3.5 DEVELOPMENT TOOLS AND ENVIRONMENTS	39
3.5.1 Microsoft Access	39
3.5.2 Balsamiq Interface Mock-ups	39

3.5.3	Microsoft Azure and SQL Management Studio.....	39
3.5.4	Appenante.....	39
3.6	CONCLUSION	39
4.	SYSTEM DESIGN AND DEVELOPMENT	41
4.1	FULFILLING THE JUST-IN-TIME PARADIGM.....	41
4.2	SITE VISIT OUTCOMES	42
4.2.1	Functional Requirements	45
4.2.2	Non-Functional Requirements	46
4.3	INTERFACE DESIGN.....	47
4.3.1	Developing User Personas.....	47
i.	Site Engineer	48
ii.	Stores Team.....	48
iii.	Team Leader.....	50
4.3.2	Creating User Stories.....	51
i.	Site Engineer	51
ii.	Stores Team.....	51
iii.	Team Leader.....	52
4.3.3	Wireframe Designs and Design Patterns.....	52
i.	Navigation	52
ii.	Capturing Form Input.....	54
4.4	APPLICATION DEVELOPMENT	56
4.4.1	Develop Overall Model	57
4.4.2	Build Feature List	59
4.4.3	Plan by Feature.....	61
4.4.4	Iterative Development: Timeboxes.....	61
i.	Timebox 1: Database Development	62
ii.	Timebox 2: Iteration 2	75
iii.	Timebox 3: Iteration 3.....	87
4.5	CONCLUSION	91
5.	SYSTEM TESTING AND EVALUATION	93
5.1	INITIAL WIREFRAME DESIGN TESTING	93
5.1.1	Heuristic Evaluation: Cognitive Walkthrough.....	93
i.	Walkthrough of Task A: Marking a Maintenance Job Complete	94
ii.	Walkthrough of Task B: Edit an Existing Component Order	100
iii.	Walkthrough of Task C: Creating a new Maintenance Job	103
5.1.2	Usability Testing	104
i.	Test Scripts	105
ii.	Participants	107
iii.	Key Findings	107
5.1.3	Summary of Wireframe Testing.....	111
5.2	PROTOTYPE MOBILE APPLICATION TESTING	111
5.2.1	Heuristic Evaluation: Cognitive Walkthrough.....	112
i.	Walkthrough of Task A:.....	112
ii.	Walkthrough of Task B:.....	115
iii.	Walkthrough of Task C:.....	116
5.2.2	Summary of Prototype Cognitive Walkthrough	117
5.3	CONCLUSION AND EVALUATION OF SYSTEM TESTING	117
6.	CONCLUSIONS AND RECOMMENDATIONS	118
6.1	PROJECT SUMMARY AND FINAL CONCLUSIONS	118
6.2	RECOMMENDATIONS FOR FURTHER RESEARCH.....	120
7.	REFLECTION.....	121
8.	REFERENCES AND CITED WORK(S).....	123
9.	BIBLIOGRAPHY AND ADDITIONAL WORK(S).....	137
10.	APPENDICES.....	140
10.1	LIST OF ACRONYMS, ABBREVIATIONS AND INITIALISMS	126
10.2	TRAITS OF AGILE METHODOLOGIES	140

10.3	DATA GOVERNANCE POLICY	146
10.3.1	Policy Purpose	146
10.3.2	Governance and Ownership.....	147
10.3.3	Data Use.....	148
10.3.4	Declaration.....	149
10.4	GWR PRIVATE COMMUNICATIONS	150
10.4.1	Scheduled Maintenance Plan Sheet	151
10.4.2	Short Term Plan (STP) Sheet: Power cars.....	152
10.4.3	Short Term Plan (STP) Sheet: Trailer Vehicles	153
10.4.4	Initial Status Sheet (ISS) Form:.....	154
10.5	SITE VISIT NOTES AND FURTHER DOCUMENTATION	142
10.5.1	WCG Reprint Costings	155
10.6	INTERFACE WIREFRAME DESIGNS	155
10.7	DATABASE SQL.....	169
10.7.1	Creating Tables.....	169
10.7.2	Creating Records.....	174
10.7.3	Queries, Triggers and Test SQL.....	176
10.8	WIREFRAME TESTING: COGNITIVE WALKTHROUGH EVALUATION OUTCOMES	180
10.8.1	Walkthrough A: Marking a Maintenance Job Complete.....	180
10.8.2	Walkthrough B: Edit an Existing Component Order	186
10.8.3	Walkthrough C: Create a new Maintenance Job	192
10.9	WIREFRAME TESTING: USABILITY TEST SCRIPTS	197
10.9.1	Participant A	197
10.9.2	Participant B	203
10.9.3	Participant C	209
10.10	WIREFRAME TESTING: SYSTEM USABILITY SCALE SCRIPTS	215
10.10.1	Participant A	215
10.10.2	Participant B	216
10.10.3	Participant C	217
10.10.4	Participant A Calculated Score Table.....	218
10.10.5	Participant B Calculated Score Table.....	218
10.10.6	Participant C Calculated Score Table	218
10.11	USER TESTING: FORMAL TEST DOCUMENTATION	220
10.11.1	User 1 Test Report.....	220
10.11.2	User 2 Test Report.....	223
10.11.3	User 3 Test Report.....	225
10.12	PROTOTYPE TESTING: COGNITIVE WALKTHROUGH EVALUATION	227
10.12.1	Walkthrough A: Marking a Maintenance Job Complete.....	227
10.12.2	Walkthrough B: Edit an Existing Component Order	235
10.12.3	Walkthrough C: Create a new Maintenance Job	241

TABLE OF FIGURES

FIGURE 1: POWER CAR NO. 43088 (FOREGROUND) PART WAY THROUGH AN EXTENSIVE 'F' EXAM	2
FIGURE 2: POWER CAR NO. 43162 HAS RECEIVED NEW WHEELSETS AS PART OF A 'G' EXAM.....	3
FIGURE 3: POWER CAR NO. 43195 HAS SUSTAINED UNFORESEEN DAMAGE. IT IS CURRENTLY BEING USED AS A SOURCE OF SPARE COMPONENTS	4
FIGURE 4: THE '5S' METHOD FOR WORKSPACE EFFICIENCY. DEVELOPED FROM [17],[25],[42].....	13
FIGURE 5: DUAL-CARD KANBAN CONFIGURATION [49],[215],[216].....	15
FIGURE 6: POSSIBLE SINGLE CARD KANBAN CONFIGURATION [49]. ADAPTED FROM [215],[216]	16
FIGURE 7: THE IMPORTANCE OF JIT VALUES FROM 1,050 STAFF WITHIN A "CONTEMPORARY" POLISH FIRM [54].....	18
FIGURE 8: USER EXPERIENCE AS A COMBINATION OF PAST EXPERIENCE, GOALS, CONTEXT AND SYSTEM DESIGN [96].....	22
FIGURE 9: MORVILLE'S UX HONEYCOMB [100],[101]	24
FIGURE 10: THE GOAL QUESTION MODEL FOR THE MEASURE FOR APPLICATION SATISFACTION [104]	25
FIGURE 11: AN EXTENSION OF GQM INVOLVING A TESTING METHOD AND ENVIRONMENT [76]	26
FIGURE 12: PLACING ACADEMIC STUDY ON THE 'RESEARCH WALL' [124]	29
FIGURE 13: DATA AS ONE OF A FIRMS MAIN ASSETS [133],[134]	33
FIGURE 14: THE CYCLICAL DESIGN AND DEVELOPMENT PROCESS PER FEATURE-LIST FEATURE [83],[84]	36
FIGURE 15: TIME AND THE NEED TO MAXIMIZE ITERATIONS	38
FIGURE 16: APPLICATION FUNCTIONALITY LIST DEVELOPED FROM A DEPOT STAFF TEAM MEETING	43
FIGURE 17: THE TWO-PANE SELECTOR PATTERN FOR PRIMARY MENU SELECTION	53
FIGURE 18: A POSSIBLE USE OF A PICKER-TYPE SPINNER.....	54
FIGURE 19: AN IOS-STYLE SWITCH HAS BEEN USED	55
FIGURE 20: SEARCHABLE LISTS FOR COMPONENT NAMES OR IDs, SIMILAR TO 'CONTACTS'	55
FIGURE 21: SEARCH BAR WITH SCOPE 'TOGGLE' BAR [181]	56
FIGURE 22: CUSTOMISABLE LIST WITH 'ADD', 'REMOVE' AND 'REORDER' IN IOS 'CONTROL CENTRE' SETTINGS.....	56
FIGURE 23: A POSSIBLE SYSTEM CLASS UML DIAGRAM	58
FIGURE 24: AN INITIAL DRAFT DATABASE SCHEMA	63
FIGURE 25: THE FINAL ENTITY RELATIONSHIP DIAGRAM (ERD) FOR THIS SCENARIO	64
FIGURE 26: AZURE ACCOUNT RESOURCES	67
FIGURE 27: DATABASE CONNECTION STRING TO ACCESS DATABASE	67
FIGURE 28: THE SQL SERVER CONNECTION WITH THE STRING PROVIDING A VALID CONNECTION TO AZURE RESOURCE	68
FIGURE 29: CONNECTED DATABASE TABLES	68
FIGURE 30: TABLE SOURCE ROW VIEW	69
FIGURE 31: A NEW COLUMN CREATED ON THE JOB_ORDER TABLE.	70
FIGURE 32: THE RETURNS RETURNED WHERE THE ACTION WAS ALTER_TABLE SHOWING THE SQL EXECUTION.....	70
FIGURE 33: TWO JOBS HAVE BEEN CARRIED OUT ON VEHICLE 43140.....	71
FIGURE 34: ALL JOBS COMPLETED AT THE FACILITY BETWEEN APRIL 1 AND APRIL 10.....	72
FIGURE 35: THE DATEDIFF FUNCTION SHOWS THREE JOBS OVER FIVE HOURS.....	73
FIGURE 36: EACH ORDER ID AND THE COUNT() OF ITS COMPONENT CONTENTS	74
FIGURE 37: DIFFERENT RECORD VIEWS.....	76
FIGURE 38: THE DATE/TIME PICKER SPINNER	77
FIGURE 39: BLANK ID TO CREATE A NEW RECORD	78
FIGURE 40: PULLING FROM THE STAFF TABLE AND THE PRIMARY KEY FIELD	79
FIGURE 41: THE STAFF ID FIELD SHOWING IDs IN EACH OF THE ROWS OF THE MAINTENANCE JOB TABLE IN THE APP AND TABLE VIEWS.....	79
FIGURE 42: SOME FIELDS OF A NEW RECORD BEFORE CLICKING "SAVE"	80
FIGURE 43: CHANGES TO THE RECORD, OR ADDITIONS TO A NEW RECORD ARE RETURNED AS SUCCESSFUL	80
FIGURE 44: VIEWING SAVED CHANGES TO, OR NEW RECORDS	81
FIGURE 45: CLICKING ON A CHANGED OR NEW RECORD TO OPEN OR DELETE THE CHANGED RECORD.....	81
FIGURE 46: AN EXAMPLE OF ONE ERRONEOUS RECORD CAPTURE	82
FIGURE 47: THIS NEW RECORD SYNC WAS PARTIALLY SUCCESSFUL	83
FIGURE 48: COMMENTS APPENDED TO JOB ID 2	84
FIGURE 49: THE CHANGE HAS SYNCED BACK TO THE APPENATE SERVER	85
FIGURE 50: THE UPDATED FORM VIEW.....	86

FIGURE 51: THE CHANGES MADE TO THE COMMENTS FIELD.....	86
FIGURE 52: THE FORM BUILDER VIEW FOR ORDER CONTENTS SCREEN	88
FIGURE 53: FOREIGN KEY-BASED TABLE LOOKUP	88
FIGURE 54: CREATING A NEW ITEM IN ORDER ID 12	90
FIGURE 55: AN INCOMPLETE UPLOAD IN THE FOREIGN KEY FIELD	91
FIGURE 56: EDITING A COMPONENT PART OF ORDER ID 12	92
FIGURE 57: EDITING AN EXISTING ORDER TO ALTER ITS COMPONENT ID	92
FIGURE 58: A SEARCHABLE TEXT BOX HAS BEEN ADDED	95
FIGURE 59: SHOWING THE KEYBOARD.....	96
FIGURE 60: A BOX NOW SHOWS CONTENTS OF SPINNER ORDER	98
FIGURE 61: A REMINDER ON THE SIGN IN SCREEN	99
FIGURE 62: SWITCHES ARE MORE SUITABLE	101
FIGURE 63: CURRENCY STEPPERS RATHER THAN SPINNERS	102
FIGURE 64: ADDING A CONTROL CAPTION	104
FIGURE 65: COST AND CHANGE ACCOMMODATION OF SOFTWARE DEVELOPMENT ACTIVITIES [63].....	142
FIGURE 66: THE POSSIBLE LONG-TERM EFFECT OF 'LEARNING' ON FUTURE PROJECTS [63].....	143
FIGURE 67: A KANBAN BOARD DESIGNED TO ALLOCATE, BALANCE AND LIMIT SOFTWARE PROJECT TASKS IN PROGRESS [217]	145

TABLE OF TABLES

TABLE 1: RESEARCH THROUGHOUT PROJECT CHAPTERS	31
TABLE 6: PROJECT TRAITS ALONGSIDE FDD AND AGILE BENEFITS	36
TABLE 7: DEVELOPER WRITE-UP OF BRAINSTORM SHEET	44
TABLE 8: FEATURE-LIST FEATURES AND GROUPED FEATURE-SETS	59
TABLE 2: DATA GOVERNANCE FRAMEWORK OVERVIEW	146
TABLE 3: POLICY GOVERNANCE AND OWNERSHIP	147
TABLE 4: DATA USE POLICY	148
TABLE 5: DATA GOVERNANCE FRAMEWORK DECLARATION	149
TABLE 9: USER A SUS SCORESHEET.....	218
TABLE 10: USER B SUS SCORESHEET.....	218
TABLE 11: USER C SUS SCORESHEET.....	219
TABLE 12: PARTICIPANT A TEST REPORT	220
TABLE 13: PARTICIPANT B TEST REPORT	223
TABLE 14: PARTICIPANT C TEST REPORT	225

1. INTRODUCTION

1.1 Case Study Processes

To develop a clear understanding of the project's aims and objectives, an understanding of the firm's current position is essential. This section will discuss the orientation of the firm, their workplace and work-practices to identify how the JIT principles impact on the firm and how the proposed application can streamline their deployment of JIT.

1.1.1 Introduction

As outlined above the project will refer to 'Great Western Railway' (GWR) [1]. Given the scope of their services across the South West of England and Wales a diverse range of trains are operated, which require various maintenance procedures. For the purpose of this project, the vehicles referred to throughout are Power cars and Trailer vehicles that form their 'Intercity 125' fleet. These vehicles cover the most intensive work and require regular maintenance at various locations. For this project site reference will refer to Laira Depot, a primary facility in Plymouth, Devon. Each maintenance facility has a finite capacity, so the timely and efficient scheduling and performance of maintenance work is essential to provide operational vehicles for all their services.

1.1.2 Existing Maintenance Processes

Maintenance performed on all stock is graded, reflecting the severity of a job. This ranges from tasks that take as little as hours (such as 'Ready to Run' or RTR Exams) right through to almost complete vehicle rebuilds that take place over a period of months ('G' Exams). The metric with which maintenance tasks are scheduled and undertaken varies by component type with Power cars seeing condition monitoring on an 'hours' basis for components such as engines, and 'mileage' for components such as wheelsets. These act as a 'flag' at which point the power car or trailer vehicle must receive an inspection and/or any resulting component changes. Most maintenance work is performed with 'packages' (containing new components for their respective grade of work) being ordered in on a JIT basis to fulfil the schedule.

A fixed planned schedule is used that outlines, based on up-to-date rolling stock component conditions and the pre-defined maintenance pre-requisites, the plan for vehicle maintenance at a facility over a period of months. Packages are ordered in and kept in on-site “stores” until they are needed.

A range of ad-hoc spreadsheet tables (often in Excel) are used to balance the planned and predicted schedule, created partly from flat-file databases containing the current vehicle ‘hours’ and mileage. Components are requested from the stores using a digital form. The processes of planning, scheduling, ordering and resource allocation is therefore not wholly centralised or synchronised. This arrangement presents the potential problem of conflicting versions due to the opening, editing and sending of such attachments via company email to staff accounts. This approach to managing a highly complex system naturally results in considerable data redundancy whereby different members of staff might need to update their version of a file on their own machine. This “*duplication of effort*” [2] is a waste of space and can lead to version confliction and similar data inconsistencies [2].



Figure 1: Power car no. 43088 (foreground) part way through an extensive 'F' exam



Figure 2: Power car *no.* 43162 has received new wheels as part of a 'G' exam

This planned schedule, or best-case scenario, is rarely perfectly fulfilled due to the corrective, unscheduled maintenance work that needs to be carried out on vehicles in response to damage or malfunction sustained in service. Such jobs are often of priority as they are being carried out on stock that is covering revenue-earning work, resulting in a reduction of vehicles available to the key planners of GWR located in Swindon, Wilts. To meet the need for spare components an array of core “strategic spares” or “stopper pieces” are kept in the stores at any one time which represent everyday items. These are replenished from time to time to maintain a certain level and cover regular jobs.

Whilst these work processes and structures are applied across the various depot sites operated by the firm, it should be noted that the physical nature of each site will conceivably fashion operational differences experienced by the user of the proposed application. It is anticipated these may include mobile-device specific matters, such as cellular signal or wireless internet coverage. This could vary widely between topographical locations.



Figure 3: Power car *no.* 43195 has sustained unforeseen damage. It is currently being used as a spares donor

The on-site computer system architecture provides security as the central information databases, operated by both the firm and Network Rail, are only accessible from the firm's network and workstation machines in the site office. For this initial system prototype it is assumed that the application must work with the internal system and not externally accessible restricted networks on which core industry data records are contained.

1.1.3 Physical Work Environment

The nature of the existing working environment has a direct impact on the use of a software application.

Natural factors such as light levels vary, and physical constraints such as the use of gloves in such a typically dirty environment will influence how a user of the application, on a given device, can perform tasks. The physical arrangement varies too, such as the position of the main site office (where administrative and work scheduling takes place) may be positioned away from the practical engineering environment. The features and design of the application, it is anticipated, will see use in both the engineering workshop and the main depot office. The on-site stores provision, handled by small site team, is also located away from both the site office and workshop in another area of the depot facility.

1.1.4 Existing Ordering and Supply Chain Processes

Components are typically ordered in advance to fulfil the planned maintenance schedule, with the level of procurement based on forecasted or projected work, a process carried out by the depot stores team. External components providers, such as UniPart Rail [3], deliver planned maintenance packages for vehicles and single, everyday parts at the request of GWR; the key factor for the firm therefore is to decide when to order which parts so that maintenance targets are met, without unnecessary financial outlay.

Short term work is harder to plan for, so the stores order short term parts based on the annual usage figures. This however is not always accurate and doesn't accommodate all un-foreseen changes. Some parts are needed at short notice and can be requested using the appropriate MRF (Material Request Form) or ISS (Initial Status Sheet).

1.1.5 Case Study Conclusions

Based on the discussion of the case study processes the developer can precisely document several key conclusions that must be taken forward into the design and development of the system. These can be written as several statements:

- Orders are placed with suppliers to fulfil a planned schedule of work
- Work towards the planned schedule is dependent on corrective and unplanned work
- Staff predominantly use flat-file databases which are shared between staff by email.
- A central rolling stock database can be used to predict work on vehicles.
- The physical workspace is distinctly different to the office space where administrative work takes place.
- The department are being pushed towards digital work practices and have recently ordered lightweight work tablets.

1.2 Project Aim

Based upon the key concepts identified in the forthcoming literature review and the existing nature of the case study firm that has been previously discussed, the following aim has been established for the project.

“To develop an application based on JIT principles to streamline rail vehicle component procurement and increase the operational efficiency of its rail vehicles.”

1.3 Project Objectives

To aid the project success evaluation, the clear definition of objectives is essential. To realise this broad aim the following objectives must be met:

1. Gain a comprehensive understanding of the development, core values and contemporary adaptations of 'Just-in-Time', e-procurement and supply chain concepts and processes.
2. Explore, through the medium of a case study, the current maintenance issues experienced in a railway firm to understand the pressures of their macro-environment in order to build an application requirement list.
3. With reference to suitable literature, design and build a mobile-based application for component e-procurement processes.
4. To understand the applications suitability following its development by evaluating the product against pre-outlined functional and non-functional requirements and its degree of usability by performing usability-based testing.
5. To clearly identify areas of the project requiring further improvement and future research.

1.4 Report Structure

Section 1.4 will outline the structure of the following report and how the sections progress and link. With the case study firm already understood, the next key foundation required for the project is the review of existing literature on the topic of JIT and mobile application usability. An extensive review of the history and development of this manufacturing and supply chain model is discussed in **section 2.2** with the history, core values, practical tools and influence discussed at length so as to highlight the important considerations for the developer in the software product.

Section 2.3 covers a broad range of topics related to mobile application use; a topic of considerable importance given the highly precise context in which the proposed solution must function. As such **section 2.3.1** looks into the value and significance of such software whilst **section 2.3.2** looks several topics surrounding the usability of mobile-based applications. This includes how they can be defined, and importantly measured so to enable a developer to understand the relative usability of their software product. To conclude this section, the author aims to acknowledge the broad topic of software usability by reflecting on how this expansive topic is itself contained within the even wider field of user experience.

The following section, **section 3**, documents the methodologies underpinning both project research and software development so that the developer can attempt to locate the project on the 'wall' of academic study. This section is of importance as it will categorise the project in order to guide the approaches used by the developer throughout in activities such as data collection, data governance and project deliverables. It also describes the range of software platforms and tools used by the developer to fulfil the different needs of the project.

The design and development of the wireframe designs, as well as the first prototype can be seen in **section 4**. This part of the document aims to, at first, understand the system users in order to identify how the software produced will be used by staff at GWR. By applying the requirements to various user's, a pathway for software development can then be layed out that suitable categorises feature development. These timeboxes are documented throughout **section 4.4.4**.

Section 5 includes all the system testing and evaluation elements that aims to understand, with particular reference to developer-level and user-based testing the suitability of the prototype system. This section links to that of **Section 6** where the developer has evaluated the application against the case study process and requirements list so as to understand how effective the application is when compared to valuable success metrics.

Section 7 gives the user the capacity to discuss the project from a personal perspective in order to highlight areas of satisfaction and dissatisfaction whilst **Section 10** contains all additional material that author has included that has particular relevance in discussions or activities carried out in the main body of the document.

2. RESEARCH AND REVIEW OF LITERATURE

Section 2 concentrates on the review of existing academic works associated with the topics of significance throughout the project. By gathering, critically reviewing and understanding their significance to the project the author can more effectively evaluate the prototype product and the methods deployed throughout the developer's assignment.

2.1 Context of Literature Review

It is important to outline how the context of the project has influenced the consultation of existing literature.

The project will develop a mobile application for UK train operating company (TOC) Great Western Railway (GWR), the operator of Inter-regional and local passenger services across the South West of England, South Wales and the Thames Valley into London. The project will focus specifically on the maintenance of the 'Intercity 125' fleet of power cars and trailer vehicles, the maintenance of which takes place at various depots. For the purpose of this case study particular reference will be made to GWR's Laira Depot maintenance facility, Plymouth.

As such the subsequent literature review will concentrate on the issues relating to supply chain management and maintenance activities, a thorough view of which one must first take in order to appreciate how traditional manufacturing concepts have seen contemporary adaptations and on what grounds they have been practiced and subsequently refined throughout history. Having highlighted supply chain management and its underlying theories, the literature will touch on the importance of supply chain continuity and, and crucially, the design and development concerns in this area. These will raise important issues that, coupled with those of usability, the developer must consider when attempting to build a mobile-based application for this purpose today.

2.2 The 'Just-in-Time' Concept

The purpose of this section is to systematically dismantle the JIT paradigm. By considering the history and conceptual foundations of this model, and the environment and economic picture of this period of

2.2.1 The History of 'Just-in-Time'

To provide a comprehensive review of the 'Just-in-Time' (JIT) manufacturing philosophy [4]–[7] and how its development transpired, it is important to first depict the state of manufacturing techniques in the motor-vehicle manufacturing industry during the 1960s and 1970s. During this period U.S manufacturing industry was largely dominant, and the popularity of a planning software, known as Material Requirements Planning (MRP) [4],[7] was widespread [8]. This software was designed to formulate achievable production processes and “*manage the inventory waste for dependant demand items*”, describing the need for individual parts or part-complete merchandise throughout the manufacturing processes, being used to forecast material quantities [4],[9],[10]. Despite its widespread use this software attracted criticism, some suggesting its rigid planning nature could not provide the necessary contract flexibility [11]. On the basis of considerable MRP uptake came 'MRP II', or Manufacturing Resource Planning [7],[12], an updated program that included a range of more generic business features to offer a more holistic view of a firm's performance [4].

At this time however different manufacturing practices in Asia, specifically Japan, were being developed [13]. The earliest beginnings of one such philosophy, Just-In-Time (JIT) [4]–[6] are widely disputed and several factors are proposed as to why manufacturing here was taking a different path to that of the United States and the West across the middle of the 20th century [13],[14]. One popular view is the lack of a domestic computer systems sector, comparable to that possessed by the United States, capable of powering the fore-mentioned MRP software. This lack of computers could have compelled the development of a more human-centric approach to manufacturing that could concentrate on actual product demand and not forecasts [14], a system possible owing to the societal benefits of a collectivist perspective. Other sources [6],[13], also recognising collectivistic culture, suggest the limited quantities of natural resources also suited the development of a new system, built on a new philosophy.

Taiichi Ohno, a former Toyota Motor Company Executive Vice-President [15], based the construction of his new system upon the production line during the 1950s to meticulously manage the business's internal operations and suppliers [16]–[18]. Ohno's personal goal was to develop a system capable of “*the total elimination of waste*” [17], seeking “*utmost rationality in the way we make things*” [15],[17]. He believed the production line, and all elements of operation, could be simplified by strictly monitoring the flow of raw materials and part-complete products into, through and out of the manufacturing system; noting that successful adoption relied heavily on the total understanding and commitment of staff in the pursuit of continual manufacturing improvement [15],[19],[20].

Given the highly versatile nature of this model, and the numerous contemporary adaptations, no single definition of Just-in-Time exists [17]. However, Ohno produced what is considered the most concise summary:

“a subsequent process should withdraw the necessary products from the preceding process in the necessary quantities at the necessary point in time”
[5],[7].

Radziwill [16], Randhawa and Singh Ahuja [23] offer alternative theories suggesting JIT in its earliest form can be seen in the development of the production line, a concept devised, practised and refined by Henry Ford in his motor-vehicle manufacturing plants across America [16],[23]. It has been argued however that Ford was not monitoring the flow of materials through the system [24]. Whilst recognising Ohno as the “father of JIT” [25],[16], Konatee *et al.* [25], suggest this model was used by Japanese shipyards to manage their stock levels. The foundations of such a model became apparent to Ohno when he visited a supermarket in America [14] and by comparing a supermarket’s business model to the aspirational elements of his new system Ohno showed the promise of this production arrangement.

“We got the idea of viewing the earlier process in a production line as a kind of store. The later process (customer) goes to the earlier process (supermarket) to acquire the needed parts (commodities) at the time and in the quantity needed. The earlier process immediately produces the quantity just taken (re-stocking the shelves)” [14],[26].

2.2.2 Concepts of ‘Just-in-Time’

Ohno believed that waste was one of the key aspects he needed to address in his philosophy in order to stand up to American competition [25], subsequently documenting seven types that could be found throughout a typical firm at this time. Waste was considered a trait of operations that placed extra, or totally unnecessary, financial or time burdens on a firm without providing added value for the customer [27].

- 1) **Transportation waste.** Moving materials between processes is wasteful. Establishing systems that minimise movement time, or the need, is vital in achieving as short a manufacturing cycle time as possible [15].

- 2) **Inventory waste.** Reducing stock levels is essential throughout the manufacturing process from raw materials to the finished product [15]. Poor inventory management typically results in an excessive inventory. Static, part or fully-complete products throughout the development process indicates precious resources consumed for its production [28]. Monczka *et al.* [27] suggest there are valid motives for holding undue materials, suitable justification being when the benefits offered overshadow the financial drawbacks of its presence [27]. Ohno appreciates this by defining ‘waste’ as resources that are both unwarranted and avoidable [29].
- 3) **Motion.** Movements of material and personnel that do not add value to the product increases costs and use valuable time. Unwarranted movements should be minimised, searching for the smallest time gains at the lowest level of production [15].
- 4) **Wait.** Stationary materials illustrate poor inter-departmental synchronisation and time that should be invested elsewhere – hence the drive to “*synchronise the workflow and operational mechanics*” [15] with techniques such as cellular manufacturing [15].
- 5) **Over-processing.** The presence of excessive production resources reflects waste conceived by the over-engineering of products [15].
- 6) **Over-production** places avoidable strain on core production and financial resources. Such practice can be found in manufacturing stages that are unnecessarily protracted, or poorly observed [15]. Superfluous stock indicates excessive time and financial outlay on unwarranted workspace, unnecessary man-hours and poorly-utilized equipment [28].
- 7) **Defects.** Defects use up precious resources that should be used to manufacture the desired quality of product [15]. Defective goods are not commercially viable and represent incoherent production management.

Based on these observations Ohno could see that these seven wasteful activities could be grouped into three main production attributes; ‘Unproductiveness’ [30], ‘Overburden’ [15] and ‘Unevenness’ [15]. It was these common production traits that Ohno knew he must precisely manage if waste was to be reduced.

Operating on a ‘Withdrawal by Subsequent Process’ basis the components move through each stage of development to exactly fulfil the required quantity at the subsequent stage [5],[28]. This directly contrasts with mass production techniques where certain time and production targets are imposed on components at each stage of production, often resulting in over-production [30],[28]. Over-production is further addressed by deploying ‘One Piece Production and Conveyance’ [5],[31] limiting

the amount of stock at, or moving from, one stage of production to create internal dependency and eliminate stagnant materials found between production stages [13],[28],[32]. Furthermore this eradicates the need for excess inventory [33], distributing work evenly across the manufacturing system to remove bottlenecks [28]. A mixed-product assembly line can now be realised as components can be called through to final assembly without delaying other production lines converging at this point [5].

Shigeo Shingo, a key player in the Toyota management team, devised a range of tools aimed at ultimately eliminating defective products [30],[19] by helping to “*avoid (yokero) inadvertent errors (poka)*” [34],[35], and with it the need for various quality control procedures [36]. Promoting the attitude “*don’t accept waste as unavoidable*” [37],[19], Shingo devised ‘Poka-Yoke’ fitted equipment to reach a Zero Quality Control state whereby the quality control process can be reduced, if not totally eliminated [36]. This equipment swaps staff for machinery in positions of vigilant or memory demanding work to reduce the margins for defect occurrence. Placing staff in more creative, less monotonous roles “*respects intelligence*” [34] and will offer greater scope to add value to the product [34]. Three main types of device are used:

- 1) Source inspection detects defects before its presence is detrimental to production, the most valuable kind [26],[38]. Machinery damage or mis-alignment prevention tools help prevent errors [26].
- 2) Machinery that signposts malfunctions whilst they are occurring, but before its presence is damaging [38]. 100% inspection actively checks for defects using tools such as a limit switch [26],[36].
- 3) Immediate action equipment stops all production when errors are identified, typically achieved with tools such as an interlocked switch [26].

The 5S method [15],[39], shown below in **Figure 4**, aims to improve the efficiency of the shop floor [40]. If followed devotedly, this linear method can bring about long-lasting change in product manufacturing. It is suggested though that some firms can struggle to go beyond the clean-up process and sustain a structured working environment [23],[39]. If successful, such changes can promote feelings of responsibility and ownership in staff, increasing productivity at the workshop level, though more direct metrics can also be applied to understand how this influenced productivity [23],[41].

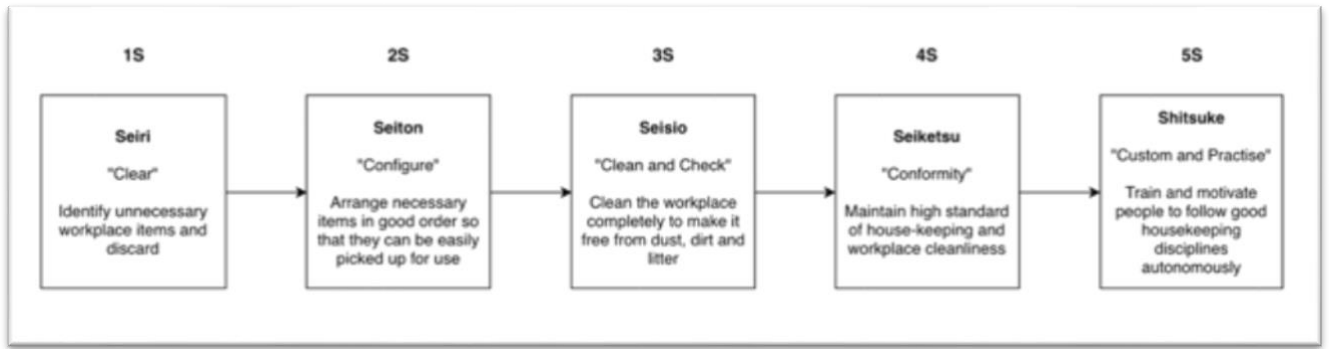


Figure 4: The '5S' method for workspace efficiency. Developed from [15],[23],[40]

The 5S steps towards efficiency form part of the much broader “Kaizen” philosophy [23],[31] a mindset that states there is always room for improvement regardless of current efficacy by concentrating on small, incremental steps to achieve better results. Nurturing the belief in staff that they can increase efficiency at their working level or location can increase efficiency, but it is key *“employee’s feeling[s], work styles and other intangible benefits should also be considered”* [41] when contemplating how to cultivate this mindset [41].

2.2.3 The Key is Kanban

Despite the significance of the tools and techniques above, Ohno recognised that levelling the production line was the real key [35]. To help restructure the stages of vehicle production at Toyota and realise a tightly scrutinized product flow he devised the concept of Kanban. This card based monitoring system, translated as *“sign board or visual signal”* [42], was designed to identify and authorise material flow and manufacturing processes [5], [42]–[46]. This ‘Pulls’ materials through the system as manufacturing capacity, and need from the subsequent stage occurs – a stark contrast to mass production techniques which ‘Push’ materials through the system regardless of production objectives or current Work-in-Process (or Work-in-Progress) (WIP) levels [45]. There are two main types of Kanban system [44],[43],[47]. The two-card approach, shown below in **Figure 5**, sees a Production Kanban and Transportation Kanban used, occasionally referred to as Conveyance and Withdraw respectively [44]. The Production Kanban is designed to specify the quantity of components or sub-components that should be manufactured by the preceding stage to fulfil material deficits at the given location [44],[47], at which point the Transportation Kanban subsequently sanctions the withdrawal of these materials from the previous stage [44],[47].

Akturk and Ehrun [44] suggest it can be difficult for businesses to jump straight to the dual-Kanban configuration, offering instead a single-card configuration. Variations of the single-card model

exist depending on the arrangement of internal processes, with physically adjacent stages seeing the removal of the Transportation Kanban, as seen in **Figure 6**, and material movement directed differently [44]. Other configurations instead retain the Transportation card and remove the Production Kanban in favour of central production planning schedulers. Kanban's power is its ability to reduce safety stock levels by reason of production levelling and strict WIP monitoring [18],[45]—this process creates an emphasis on material quality that helps produce higher quality products with considerably reduced waste.

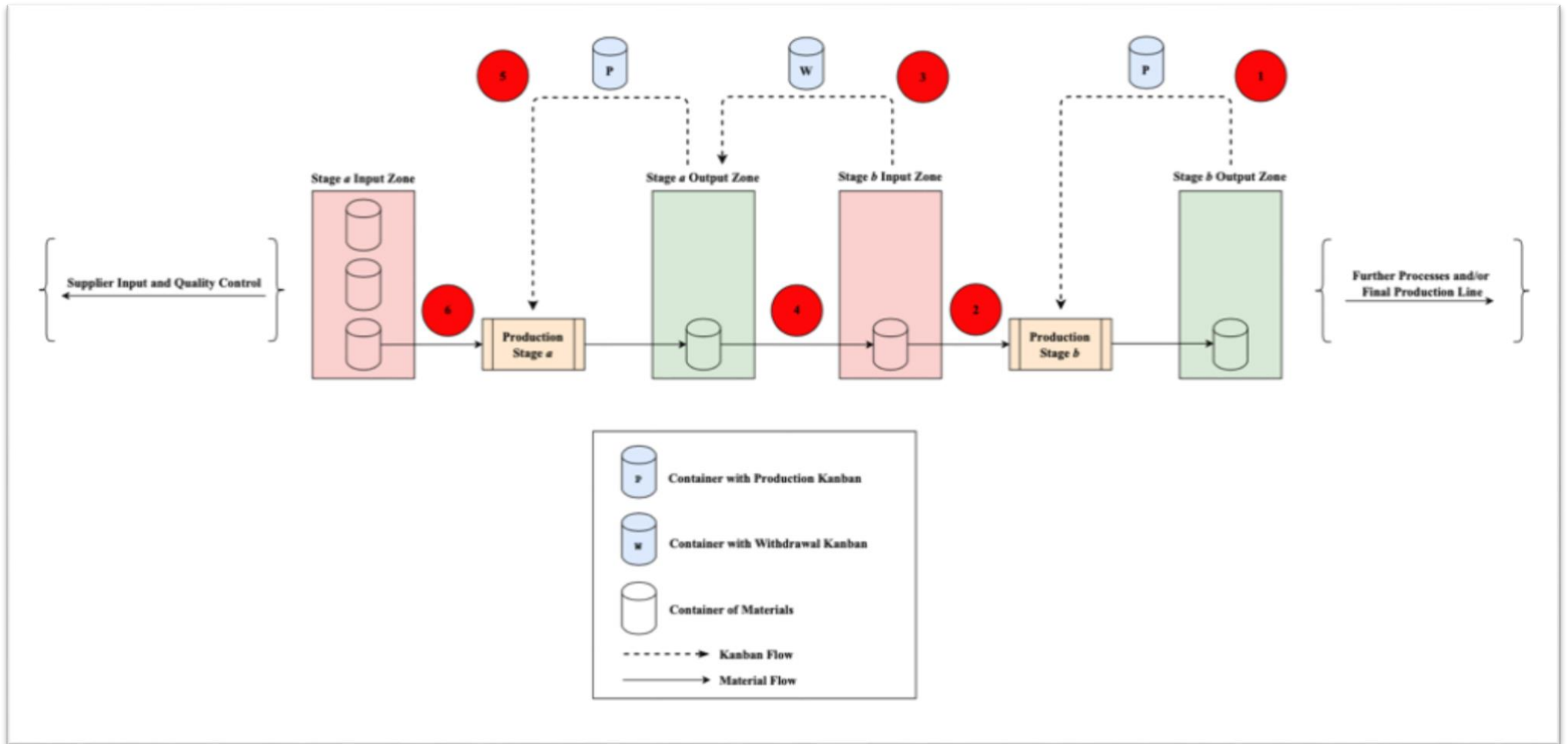


Figure 5: Dual-card Kanban configuration [47],[209],[210]

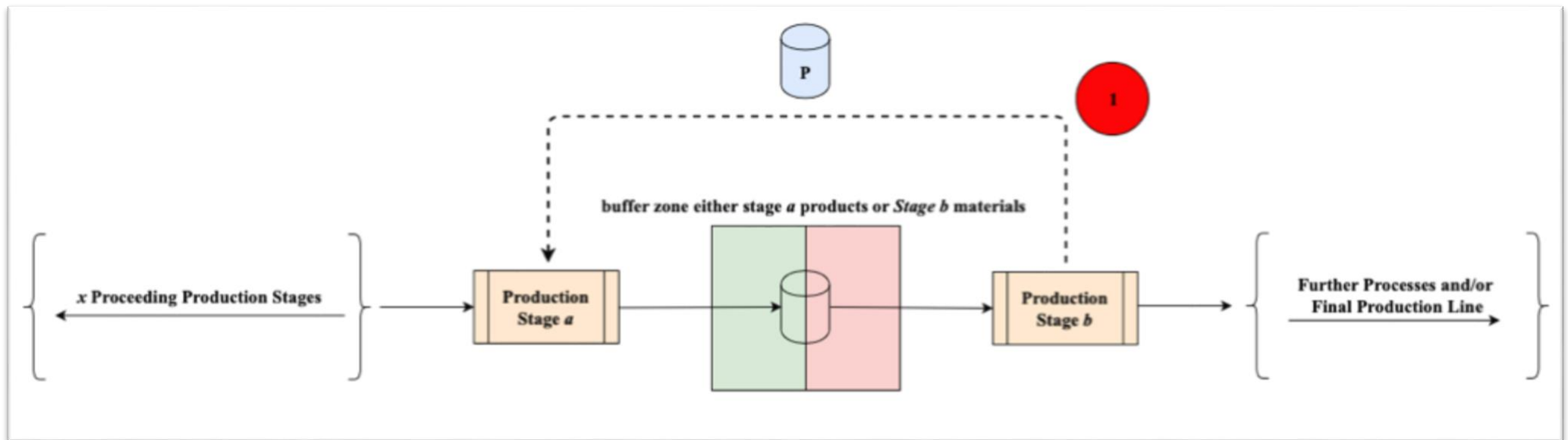


Figure 6: Possible single card Kanban configuration [47]. Adapted from [209],[210]

2.2.4 Influence and Uptake

Following comprehensive realisation at Toyota during the last three-decades of the 20th century the recognition of JIT grew globally, and its results were enviable [14],[18]. Across the 1970s and '80s similar vehicle manufacturers in America took up the philosophy. By not paying suitably close attention to its definition, or the monitoring of internal structures and employee mindset, the results of the new system were believed to be limited [13],[15],[27],[41]. The publication of Womack *et al.*'s [18] landmark paper, 'The Machine that Changed the World' was intended to educate Western manufacturing firms, focusing on the tools and techniques fundamental to the Toyota Production System (TPS) [49] to provide actionable steps [16]. However, according to Liker [13] in his 2004 publication, JIT was still causing considerable issues. He believed that the hugely broad management factors of JIT, overlooked in that piece, had as much significance as the precise techniques with which to reduce waste [13],[32],[50].

To exhibit the belief that core values were being disregarded, Liker [13] refers to the work of the 'Toyota Supplier Support Centre' (TSSC) [13]. This advisory group, led by Hajime Ohba, was set up to help educate American businesses that were struggling to implement lean practices in their business model [13]. To help illustrate this lack of understanding the TSSC partnered an anonymous US business – which agreed to provide the TSSC access to one of its production lines – to conduct a nine-month study of its operations. The study eventually achieved the following improvements [13].

- Lead times were nearly slashed in half - *"46% reduction in lead-time to produce the product (from 12 to 6.5 hours)"* [13].
- A considerable reduction of the WIP level *"from 9 to 1.5 hours"* of 83% [13], critical in achieving a level workflow.
- The line boasted a truly responsive inventory with a 91% reduction in finished-goods inventory. Complete products fell from *"30,500 to 2,890 units"* over the study period [13].
- Staff overtime was halved. *"10 hours to 5 hours/person-week"* [13].

The benefits of JIT at a strategic level can be clearly seen in the case study as this award-winning firm, considered by many to be highly proficient, still had sizeable scope for improvement [13]. These performance figures successfully showed how a clear focus on JIT's core values carries arguably more significance than practical tools to reduce production waste, a widely supported theory [35],[51],[50] aided by a 2015 study by Grzelczak and Werner-Lewandowska [52]. In their survey of 1200 employees from a "contemporary" [52] Polish company, the JIT principles 'priority of value added', 'standardisation' and 'feedback' were labelled as "not relevant" [52] or of little importance by

the 1050 random responses. Nearly half the survey correspondents considered ‘feedback’ to be of “small importance” [52] with approximately ten percent labelling it as “not relevant at all” [52]—a far cry from Ohno’s original and integral elements of transparency and continual learning.

The full table of responses from this study is shown below in **Figure 7**.

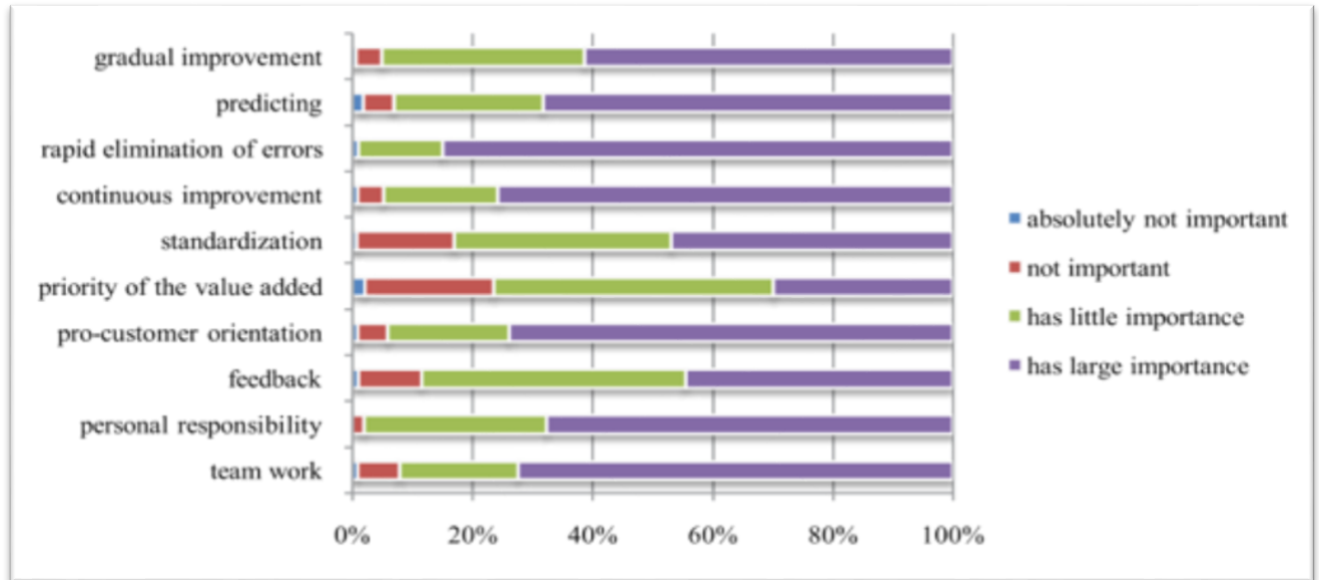


Figure 7: The importance of JIT values from 1,050 staff within a “contemporary” Polish firm [52]

2.2.5 ‘Just-in-Time’ Supply Chain Management

It can be clearly seen that the JIT model concentrates on internal improvement, yet firms must understand how numerous external factors can dramatically shape the delicate internal model. The scope of JIT can be widened to encapsulate the entire production system and critically help regulate external forces, a process known as Supply Chain Management (SCM) [53]–[55]. SCM concentrates on inspecting material flow and the efficient coordination of production tasks, making it a broader area than just logistics believes Christopher [54]. Zimmer [53] recognises how the perspective with which “information, materials, capital and manpower” [53] shifts from the internal JIT perspective to one that provides “linkage and coordination” [54] between the various activities carried out by all entities in the production cycle [54], a viewpoint supported by Oliver and Webber [56] (where an entity is a raw material provider, an end customer, any other intermediaries, and the firm itself [50],[51],[53]). The holistic view of the manufacturing processes found in “virtual enterprises” [57],[58] that such systems provide aims to reduce supply chain deficiencies [54] can be compared to those initially documented by Ohno [27]. Vokurka and Lummus [24] note these similarities stating that many JIT-

oriented firms therefore already have the necessary fundamentals to practise this wider concept, but like the implementation issues experienced for JIT, management appreciation is paramount [24],[50].

Analytical Supply Chain Management (ASCM) systems amalgamate various planning and management factors concurrently, allowing 'real-time' amendments to the ordering, movement and consumption of materials [4]. Such applications are typically nested in Enterprise Resource Planning (ERP) platforms [58],[59]. Given the nature of some large firms however, an integrated solution is not always viable – such scope instead suiting independent systems, a decision driven by the need to deliver competitive advantage [58]. The highly variable nature of ERP systems, and the option to tailor service functionality through interfaces connections with various 'Application Program Interfaces' (APIs) [60], helps cater for variable market conditions, changing business relationships whilst providing suitable 'goal-orientation' (the process of managing goals set by supply chain entities to provide both workflow integration and integrity), to improve the efficiency of the chain as well as its components [52],[54],[55],[57]. Spekman [61] observes the significance of organisational goals, a factor that (along with strategic misalignment), is noted as a common cause for the *"disintegration of [supply chain] networks"* [61].

Pavnaskar, Gershenson and Jambekar [19] highlight numerous areas an integrated JIT supply chain can benefit a company, though in contrast to the internal JIT metrics with which to gauge performance they are now predominantly time-based to reflect how the relationship with other supply chain entities influences their performance [54].

- Defect reductions of up to 20% per year
 - up to a 50% quality improvement
- 80% reduction in shop floor space
 - by virtue of inventory reductions of greater than 75%.
- Cumulative lead time, the time difference between receiving an order and the completion of the finished product [7], reduced by more than 75% [19].

Implementing all the preceding aspects of the JIT philosophy means that you have reached the state of 'Jidoka' [28]. Ohno's end point was not a given level of efficiency but rather a position of 'automation with the human touch', making Jidoka one of the numerous practical tools helping to enforce the quality mindset, Kaizen, the pursuit for continual efficiency improvement [13]. Practically Kanban too plays a crucial part in the philosophy, stressing that *"people know what to do; they don't*

always do what they know. By creating clarity on the effects of their actions, Kanban encourages people to do what they know” [62].

2.3 Designing Mobile Applications

2.3.1 Uptake and Value of Mobile Devices

Mobile application development does not possess a rich history [63],[64]. As late as 2008 the AppStore for iPhone was released by Apple, starting the mass production of mobile devices and encouraging the widespread development of mobile software [77],[79],[80]. Just two years after the introduction of Apple’s store it was estimated there were approximately 250,000 applications available through Apple and their competitors, several of which subsequently launched their own stores [63],[66]. Just five years later, it was estimated that global app store downloads were at approximately 102 million with projected figures during 2013 forecasting over 260 million downloads by 2017 [66]. According to a 2016 study, the number of mobile app downloads was estimated to have totalled 25 billion on Apple’s iOS platform alone [67].

Tablet computers, relatively obscure until the release of Apple’s iPad during 2010 [68] thanks to their as yet *“extremely niche markets”* [69], helped give rise to another clearly distinctive category [70] of ‘mobile’ device from the existing phones and ‘Personal Digital Assistants’ (PDAs) that had previously existed [68], [70]. A larger screen that *“integrate[s] input by touch and output display”* [68] compared to that of a phone or PDA, presents several notable design nuances [69]. This larger physical composition represents an obvious extension to a desktop or laptop computer for businesses, making them more common in places revolving around regular employee movement and relocation—changing the traditional fixed working space [70]. Their lightness, dexterity and inherent portability are particularly attractive to staff where the relative centralisation of a work environment is blurred or indistinct. As highlighted above and explored later in the design of an application for GWR, the physical separation between engineering work and office-based management and administration at GWR’s Laira Depot is clearly defined, offering a suitable compromise between the office desktop and a company smartphone.

2.3.2 Usability of Mobile Applications

i. Defining Usability

Evidently the significance of mobile applications in a business environment is growing [71],[72]. The developer has a range of various design and development concerns, several of which are mobile specific [63]. At the heart of application design, irrespective of platform, are the issues of usability and user experience, a family of key tools in gauging the suitability and overall success of a software platform; a concept far more than purely ease of application use [73]–[75]. The study of usability aims to evaluate the extent, and relative success of, a user(s) interaction(s) with the application in its native environment [76]–[78] - a broad subject that is hard to define but easy to appreciate when its consideration is omitted suggest Rubin and Chisnell (insinuating this process should take place during product design [79],[80]). Given the broad nature of usability and extensive research no single definition prevails - though various standardisation attempts have been made. One such accepted definition is offered by ISO 9241-11, focusing on the definition of ‘successful’ software use [81]–[83].

“The extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” [65],[77],[84]

ISO FDIS 9241-210 [85],[80] standard comes at usability from the user experience perspective – considering user experience an extension of the metrics associated with usability – despite relative ambiguity in pronouncing whether usability is actually part of the user experience field. The following definition for user experience is offered [85]:

“A person’s perceptions and responses that result from the use and/or anticipated use of a product, system or service” [85],[86],[87]

ii. Usability in User Experience

Based partly on definitions provided by ISO FDIS 9241-210 and the like that recognise usability as a section of User Experience (UX), the consultation of relevant metrics and models for interface evaluation is widely recognised [88]. Such models fall into either ‘top-down’ and ‘bottom-up’ arrangements that concentrate on a user’s perspective [80], by measuring goal fulfilment, or design, by inspecting the success of interface components from the developers viewpoint [86]. Bevan [83] considers these essential - suggesting system scope, complexity and context changeability have an effect on software usage, meaning quality cannot be directly considered a software attribute [83]. McCarthy and Wright [89] support this viewpoint offering a relational theory

that notes how interactions with the systems interface, use in its designated context and recollected past feelings influence forthcoming experiences [86],[87]. A visual representation, shown in **Figure 8**, was developed by Kankainen [90] and discussed in a web context by Maehr *et al.* [91]. These foundations are supported by Nielsen [92], who suggests how the variable nature of a software environment influences user response and ease of problem identification; ultimately emphasizing the difficulty in finding all usability discrepancies throughout a system.

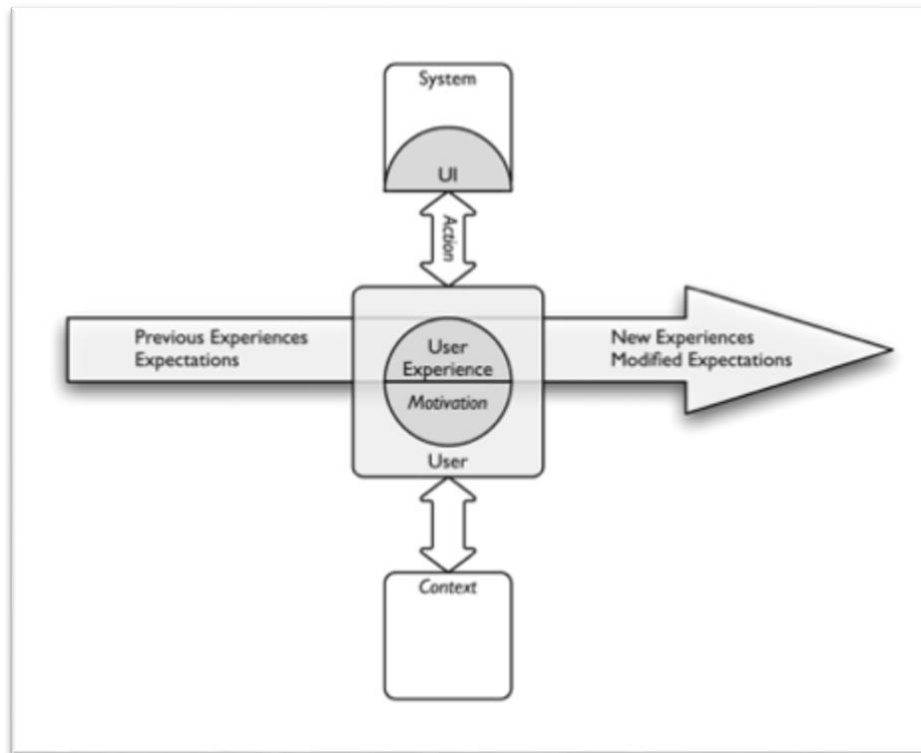


Figure 8: User Experience as a combination of past experience, goals, context and system design [91]

The ISO 9241 standard [81]–[83], introduced above, attempts to define the relative degree of software usability by placing various Quality of Use metrics against the application in its native context, the outcomes offering a value of usability – a comparable method to that offered by Nielsen [93] and his 5 key “*quality components*” [93]. Aimed at design which offer hardware and software propositions, this measure of usability concentrates on three main elements [86],[94].

- Effectiveness is a very broad term but is frequently measured by evaluating the extent to which a user’s goals are realised [94],[86].
- Efficiency reviews the resources used in the process of goal fulfilment, such as memory load, time expense or financial outlay [94],[86].

- Satisfaction refers to the users feelings whilst carrying out tasks associated with goal realisation [94],[86].

Alongside precise usability standardisation attempts sit more holistic tools to help evaluate the usefulness of a system. Figure 9. illustrates Morville's [95],[96] honeycomb framework, a graphic that appreciates the importance of usability, but moreover how it should be considered only part of an effective software solution [95].

- **Useful:** Software must fulfil its intended purpose [97]
- **Usable:** The system should be straightforward to pick up and learn and easy to use in its context [97]
- **Desirable:** Must consider elements of emotional design and the consideration of corporate identity, as well as being easily translatable [95],[97].
- **Findable:** This is a highly variable software attribute and is not always of significance. Wesolko [97], referring to Morville's work, considers this as information contained within the system, rather than a quality of the software. This implies that the underlying information architecture impacts on the success of information retrieval [97].
- **Accessible:** An application embraces accessible design guidelines that offer inclusion and provides continuity [78],[95].
- **Credible:** The extent to which a solution is trusted in its given role by a particular user [78],[95].

When all of the applicable facets are appropriately met it is deemed that *valuable* software has been produced. Morville [95] does note however that software context influences the extent to which the different elements are important for a given software product, making this model as much a design categorisation tool as a UX component framework [78].



Figure 9: Morville's UX honeycomb [95],[96]

The honeycomb [95], later widened by Revang in his UX wheel [98], acts somewhat as a starting point for the Goal Question Metric model [99]. This model was not developed specifically as an evaluative tool, but more a quality improvement structure whereby usability evaluation questions are developed for each key software attribute. Suitable design metrics and testing elements can consequently be established for each usability measure, such as that of user “Satisfaction”, shown in Figure 10[99]. This model itself has been further extended by Zhang and Adipat [71] to include the clear definition of the testing methodology and environment and so is of particular use during the usability testing stage, as illustrated below in Figure 11. Alongside the Honeycomb method to evaluate the significant software attribute, tools can be used to evaluate where usability decisions will influence software. Van Greunen, Van der Merwe and Kotze [78] directly link UX and HCI design considerations to the applications requirements list, believing that a project’s functional and non-functional requirements can be placed against the different areas of a business process management tool to see more clearly where specific design decisions, people, business and technological elements of new software are of particular value [78].

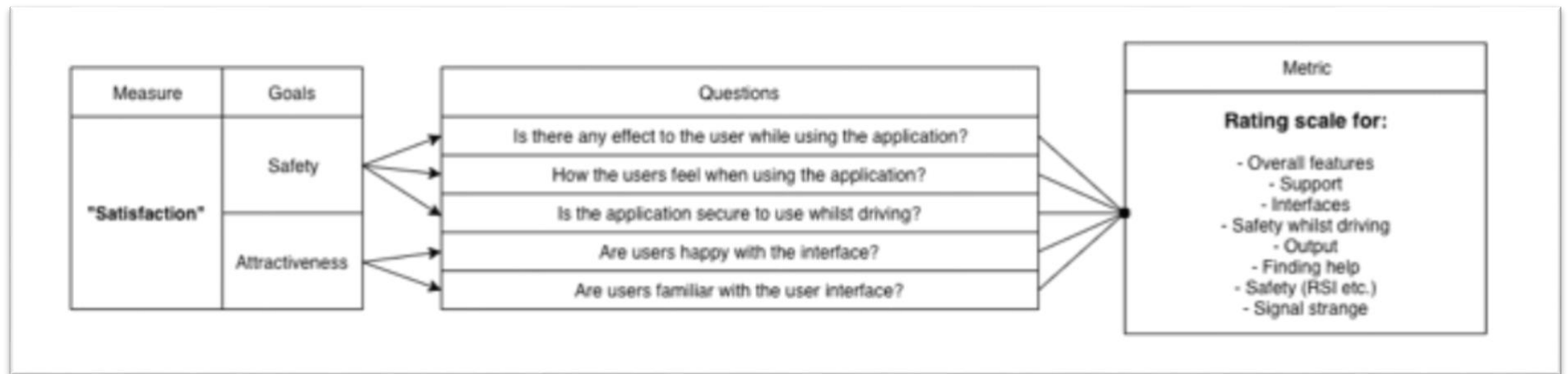


Figure 10: The Goal Question Model for the measure for application satisfaction [99]

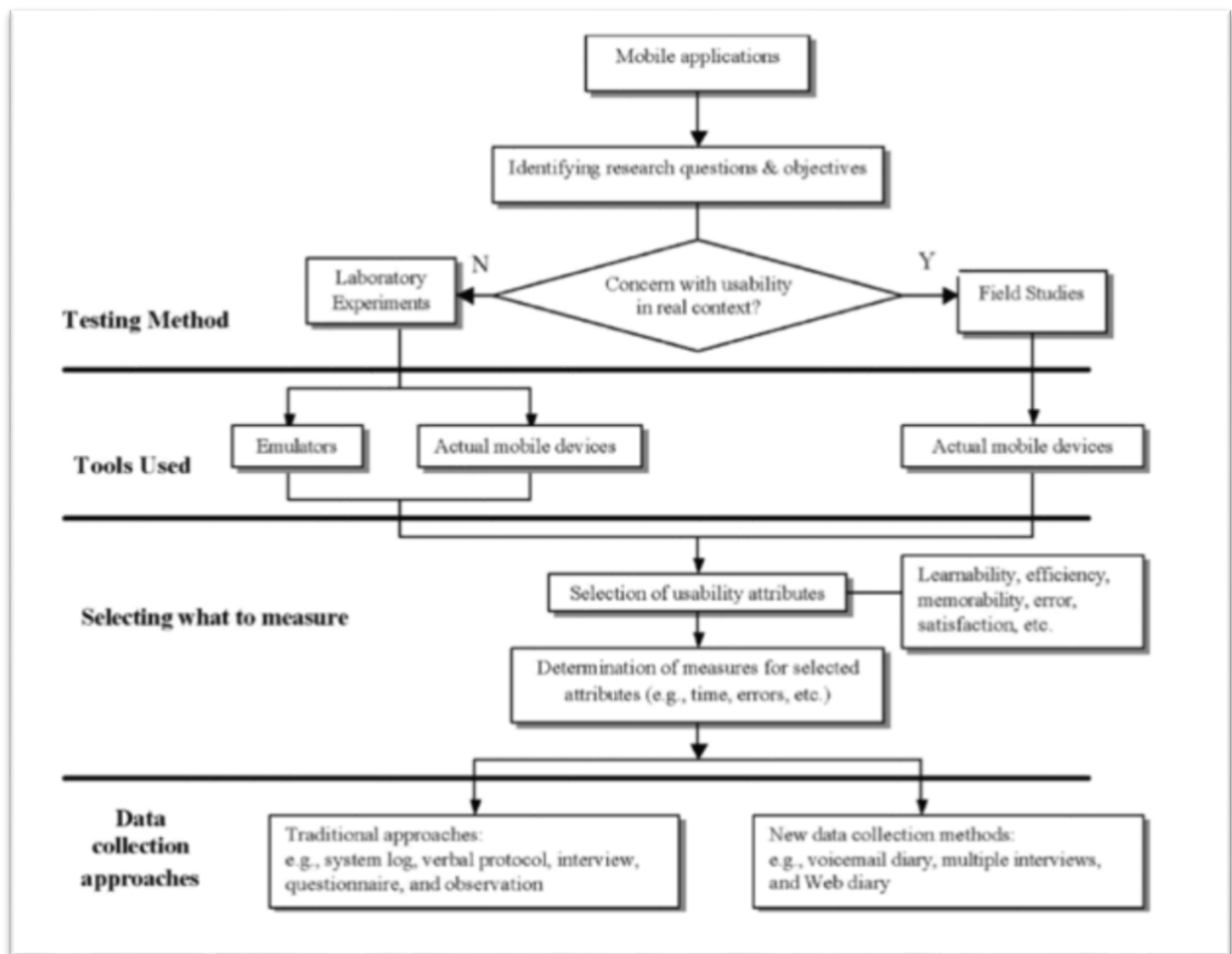


Figure 11: An extension of GQM involving a testing method and environment [71]

Whilst the issue of software usability is chronic, mobile as a platform, by cause of its portable nature, offers more specific design, development and implementation challenges [100], many of which position themselves in the void between device self-sufficiency and system interdependence [100].

- The physical attributes of mobile devices, namely the smaller screen size and very different input techniques, require different design approaches [73],[71],[101],[102],[103]. The icons and pointers used in desktop or laptop computers are now no longer suitable and require redesigning [104].
- Further exacerbating the issue is the large variety of mobile hardware and software, so “*a generic design for all platforms does not exist*” [105]. Such disparities conceal behavioural inconsistencies because design regularity is no longer feasible [105].

- Computational capacity: Memory and processing capability of mobile devices is typically limited, restricting both the range of usable functions open to the developer and the contexts within which mobile power is sufficient [71],[101],[102].
- Gong and Tarasewich [101] and Schmidt [106] suggest that the mobile environment often places greater time pressure on activity performance, often demanding the time reduction for core or recurring functionality should be accomplished, requiring the use of tested HCI concepts.
- The physical composition of mobile devices influences the complexity of testing [63],[107],[108],[109] frameworks and methodologies.

Context presents several differing design and usability concerns. As part of the much wider Human-Computer Interaction (HCI) field [104],[108]–[110], its significance being the examination of the various triggers, “... *[a] person, place or object that is considered relevant to the interaction between a user and an application*” [113],[114] that influence device response and, as such, the consequential interaction design choices open to the developer [110],[112],[114]. There are several key types of context that dictate how suitable interaction can be ‘shaped’ in a system [86],[115],[103].

- **Infrastructure Context:** Mobile systems are likely to be used in different infrastructure settings. How the device responds to the information system infrastructure at a given location is crucial in defining the suitability of the solution in its given role [111], especially in safety systems where dependability is a key system attribute, an issue that runs significantly deeper than the interface [111].
- **System Context:** The complexity of the underlying system influences the design decisions made by the designer. The use of distributed systems is widespread, the precise back-end configuration of which dictates the design choices available to the application and interface designer [111].
- **Location Context:** The type of device used and its primary form of interaction used is no longer the sole factor in the field of HCI [111]. An understanding of the location, and its attributes, in which the system will be used must be gained. How the system interacts with the device at a given location will influence the interaction techniques built on top of the underlying system [111]. The physical environment, the location of the app and its users will influence the content displayed, if not the underlying information architecture.
- **Physical Context:** The nature of the physical environment influences how the application will be developed. Hardware, a context in which the system will fit, provides various modules with

which to develop an application, influencing how an application can operate and how it can be used by individual(s) [111]. Different design choices can now be made based on the various operational constraints imposed by the physical device - screen size being a typical physical factor [111].

3. RESEARCH AND DEVELOPMENT METHODOLOGIES

This section aims to provide a comprehensive project definition whereby the projects research perspectives, precise development methodologies and resource usage are outlined and suitably justified so as to ensure a systematic project has been undertaken [116]. To provide a realistic conceptual design the project foundations and scope must be appropriately noted, a task fulfilled by undertaking more precise technical design research [117]. This aims to suitably document the project's research perspectives and the development methodology used so as to ensure a clear project outline and viability. As such it is more precise than the wider scope as it considers how the broader conceptual design defined can be met [117].

3.1 Research Methodology

In order to place this project in the appropriate section of academic study the use of the 'research wall', as illustrated below in **Figure 12**, is used to aid the development of a research framework [117]. This framework requires the definition of a philosophy, a project approach and several project strategies whereby the wall is split vertically indicating development and survey-based study. As highlighted below this project should be considered relatively central, but just leaning towards a development-based research project.

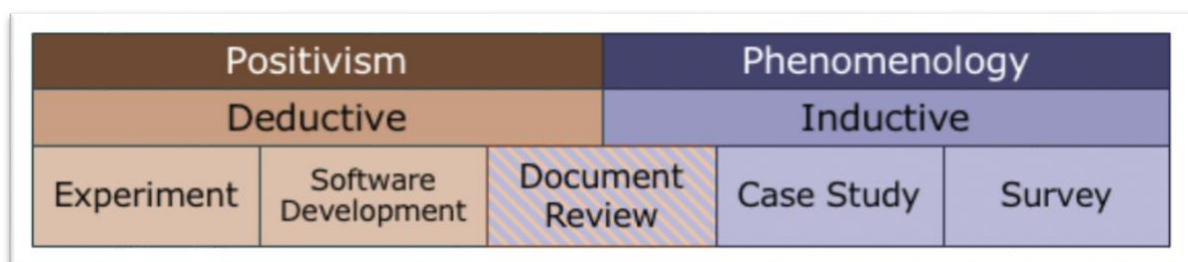


Figure 12: Placing academic study on the 'research wall' [118]

3.1.1 Research Philosophy

Owing to the design, development and testing of mobile software, this project most closely aligns with a positivist perspective. Despite several more abstract usability tools influencing this development that are not statistically grounded, this project will provide direct frameworks aiding the construction of a tool that, by means of other theoretical metrics, can be measured precisely and on which *"objective facts and established truths"* [119] can be offered. This distinct difference is noted by

Usher and Whitty [117] in their reference to the contrast between clear project success and project satisfaction. The former is based upon such truths that the latter, thanks to the ambiguous nature of 'value' perception, is far less reliable in a technically-founded process. The fields of Usability and User Experience (UX hereafter), the subjective and qualitative experiences of a user when interacting with a system, are still based on quantitative foundations built up over time [117].

3.1.2 Research Approach

In terms of approach the project is clearly developmental as the nature of the case study and software concepts applied are constructed around a number of key statements regarding their intended use, often in the format of a project goal which can then be observed and/or tested during the project lifecycle [120]. In the development of a 'real-world' software problem such as this the process of product development revolves around a clear and defined starting point of academic research. The acts of applying, discussing and testing such understood or pre-emptive statements in relation to this scenario can be more accurately assessed by a review existing literature that aims to highlight and discuss these as appropriate.

3.1.3 Research Strategy

To appropriately triangulate a response to the research propositions, and evaluate their suitability for the project, three strategies will be used. The process of results triangulation helps provide a foundation for a considered evaluation of the research title ensuring that a strong set of project outcomes can be delivered at the project's conclusion.

Primarily, an extensive literature review will be undertaken that will involve systematically evaluating existing research in the field of JIT manufacturing and supply chain management to underline the key concepts, works and authors central to this manufacturing management technique.

Alongside the review of Just-in-Time, the concepts and issues surrounding software development evaluation metrics, fundamental usability will also be explored; a process that aims to provide a basis to more accurately understand the context of the second strategy; the issues faced by a specific industrial case study. This partner will be a contemporary industrial firm so as to understand the pressures on their business and the particular software needs of a firm feasibly looking to implement new software.

Based on reference to the academic material and case study scenario, a software application will be developed that will aim to support the case study processes in such a way that aligns with both the core JIT values and the key software development methodologies and concepts (through which project success can be gauged).

3.2 Data Collection

3.2.1 Collection Methods

The process of data collection throughout the project is vital in developing a system that aligns with the systems currently used by the firm. The use of prototypical data and ‘real world’ nature of information contained within the system aids in the design of a system that can be “useful” in relation to the Morville honeycomb, discussed in **section 2.3.2**, and for the end customer.

- Secondary data in the form of spreadsheet and table views from the industry partner represent quantitative data thanks to their directly quantifiable results. These administrative communications can be seen in full in **section 1.1**.
- Secondary qualitative data from the firm regarding the usability of the application and its relative usefulness. This softer data is of considerable value as a requirement-gathering technique and is discussed at greater length in **section 3.2.2**.
- An Evaluation against values and concepts of a supply chain methodology using data collated during the primary review of existing literature, a process that involves the collection of secondary qualitative data from various academic repositories.

The nature and location of data collection throughout the project is shown below in table 1. *Chapters without pre-defined data collection, or those that use existing material(s) are omitted for clarity.*

Table 1: Research throughout project chapters

Report Section	Research Perspective
The Just-In-Time Concept Chapter 2.3	Qualitative – literature review Action Research – considering concepts and understanding to establish conclusions based on existing academic material(s)
Designing Mobile Applications Chapter 2.4	Qualitative – literature review Action Research – considering concepts and understanding to establish conclusions based on existing academic material(s)
Development Methodologies Chapter 3.4	Action Research – Reflecting upon literature review findings to define suitable model

Site Visit Outcomes <i>Chapter 4.2</i>	Ethnographic Case Study – Site visits, Observations and informal Questionnaires Archival – Reference to current and historical internal case study firm private communications Action Research – Reflecting upon key elements to findings to establish development considerations.
Conclusions and Recommendations <i>Chapter 6</i>	Action Research – Reflecting upon findings and establishing understanding to recommend further work(s)

3.2.2 Requirements Elicitation Processes

An attribute and recognised benefit of a rigorously constructed requirements list is that it is suitably prioritised in a way that defines the order of software features that meet with client approval. This process sees the more important features delivered in the first development iterations with those with less significance designed, developed and integrated into the solution later in the development process [121]. The task of requirement prioritisation is most successful when the client and developer actively liaise in order to create a list that both closely aligns with the time and functionality needs of the customer and that provides the necessary technical prioritisation for the developer [121].

Glinz [122], references ISO 9000:2000 [123] suggesting that all requirements, both functional and non-functional, emphasize software ‘quality’ given its ability to illustrate the “*degree to which a set of inherent characteristics fulfils requirements*” [123]. This notion is most clearly seen in the system’s non-functional requirements, as they help describe the more holistic demands of the system thanks to their non-specific nature.

The developer has undertaken two site visits to the Plymouth-based maintenance facility, meeting with a range of site staff with the aim of gathering a broad range of requirements, opinions and preferences associated with the development of a maintenance support tool. Comprehensive notes were made in response to specific questions asked by the project individual as well as comments and observations related to the design and development of such a tool for the site team. Taking several hours each time, the informal results of the staff brainstorm meeting and subsequent interviews and discussions held with various members of staff are shown in **Figure 16**.

To help collect a combination of broad and precise requirements, opinions and preferences, several methods were deployed. Interviews are a fundamental elicitation technique designed to collect views held by both potential users of a system and other central stakeholders. This process helps to collate a broad range of information thanks to the conversational nature such open interviews can present. This allows the interviewer to adjust questioning accordingly in order to explore

additional paths aimed at furthering their understanding of existing systems and the needs of the software product to be developed [124].

Brainstorming too is common and is of particular value in *“help[ing] to develop creative solutions to specific problems”* [124], a process of further benefit when this can be undertaken with the client to agree on precise elements of software features that present detailed system elements [124]. Brainstorming is of particular value in helping to tease out and understand how to satisfy issues currently being faced by the firm – an even more valuable process when performed as a group. This particular requirement-gathering techniques was used heavily due to *“a limited understanding of the domain”* [125] and helped to highlight areas of more direct follow-up questioning.

3.2.3 Data Governance

When conducting a project that relies upon peripheral data it is important to understand the various pressures on data usage, storage, access and publication. To recognise this, Section 3 will briefly discuss the importance of digital data in this project. Information transparency is an element of considerable significance given the premise of *“data as an asset”* [126] and its importance to firms in this information-centric age. This understanding is illustrated visually in an adapted version of Weill and Ross’ representation of key business assets and the weight of digital information [127].

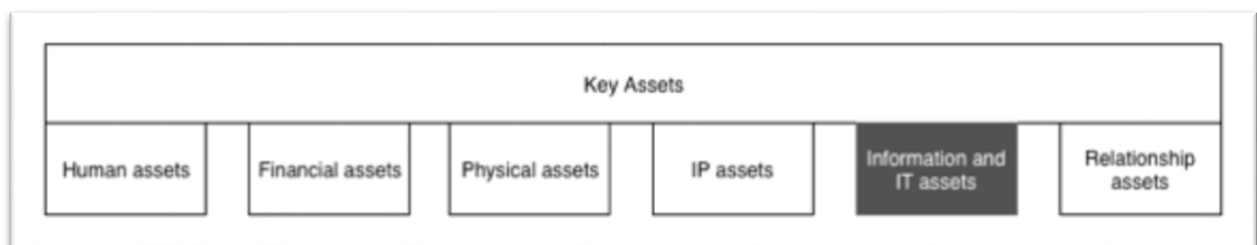


Figure 13: Data as one of a firms main assets [126],[127]

A comprehensive policy aims to ensure a high-level of data quality; that is the degree to which data is up to date, accessible and consistent with business state and purpose [128].

The aim of ensuring a high degree of data quality is built around the understanding that there are many issues experienced when working with corporate data:

- **Data needs to be understood:** data must be understood for its worth to be established, so somebody needs to know if it is good and if it is of use [129].
- **Data is shared:** lots of people typically use data so it is critical to outline how it can be shared and who is responsible for any issues that occur to data after its creation.

- **Software development often requires handoffs:** various development models require data movement that presents a possible threat to its security, integrity and continuity.
- **Blurred Accountability:** *“Processes alone do not ensure the integrity of information”* [130]; hence the value in such policies to clearly outline responsibilities and processes.

Appendix 10.2 outlines data purpose, access, alteration and the responsibilities of the developer throughout the project to ensure appropriate use of obtained data.

3.3 Project Deliverables

The deliverables for this project will be founded on research into various subjects including:

- Just-in-Time and associated literature
- a suitable software development paradigm
- the existing processes and pressures experienced by an industry partner
- software application usability

Beyond the theoretical outcomes an initial prototype system will be developed and presented to the client which they can use, or reference as deemed appropriate. The selection of an Agile software development methodology is important as such paradigms consider the state of project deliverables to be a definition of project progress and success [131]. Alongside the prototype solution, a written report documenting both the fore-mentioned areas of literature and the relevant decisions made by the project developer will be produced.

3.4 Development Methodology

3.4.1 Agile Software Development

‘Agile’ is a software development philosophy that shares similar goals to that of JIT. Its aim *“to promote quick response to changing environments, changes in user requirements, accelerated project deadlines and the like”* [132],[133],[134] is supported by Conboy and Fitzgerald [135] who identified the Toyota Production System, introduced in **section 2.2.4** as a foundation for this family of software development methodologies. The 2001 Agile Manifesto developed by the Agile Alliance [136]–[140] aimed to document various agile elements that firms (either fully or partly) had deployed in an impromptu manner up until this time, many firms believing that they were at the pinnacle of efficiency [141],[142]. A definitive definition of ‘agile’ methodologies is hard to provide due to the highly variable

nature of the term [135],[142],[143], further supporting the common belief that these methodologies, in view of their nature, are not suitable for all projects [138],[144],[145].

3.4.2 Feature-Driven Development

Given the wide-ranging values and broad guidelines provided by the Manifesto, many precise methodologies have been developed [143] that comfortably align with the 12 Agile points. There are few tools with which to select the best paradigm offering the suitable degree of agility. This selection process demands the precise comparison of methodologies to identify concepts suiting the given development scenario [144].

i. Concepts of Feature-Driven Development

One prevalent method is Feature Driven Development (FDD) [143],[144], a natural evolution of the 'Coad Method' [144]. This model suggests that development comprises two main stages, the creation of a features list and the process of 'feature-by-feature' development [144],[146]. Chowdhury and Huda [144] note that the quality and accompanying attributes of project code constructed during the development stage is dependent on the value of the features list, requirement documentation, UML diagrams and work packages. The methodology consists of five key tasks [147],[148], starting with an evaluation of the software's requirements for each problem in the software's destination field, whereby a requirement is a deliverable, "*client-valued*" [147], box of product functionality. The result of this is a list of precise functionalities, collated to form feature sets [147]. Based on technical and time pre-requisites the list of feature sets and functionalities can be appropriately time-boxed to identify their appropriate position in the development process, the position being equally influenced by the clients perspective on priorities [147]. Classes comprising each feature set are allocated to developers inside 'feature teams' [147] to ensure precise code responsibilities and ownership, an essential step given the significance placed on developers in an agile environment. The final process combines both design and development activities for each feature-list feature, as seen in **Figure 14**. [145]. This process is aided by consulting design and development tools to produce working classes and methods. The development of existing documentation, and frequent testing processes, makes the task of feature amalgamation more straightforward [147].

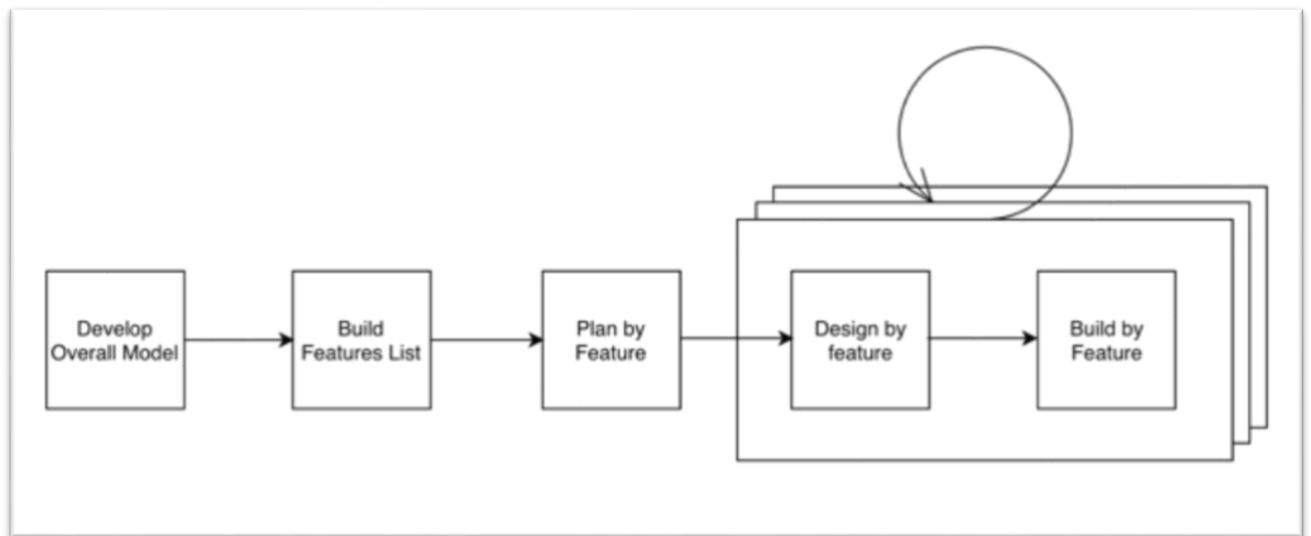


Figure 14: The cyclical design and development process per feature-list feature [83],[84]

ii. Project Scenario and Adapting FDD

This section aims to understand how the choice of an Agile methodology is suitable for this project. To do this the traits of Agile, and FDD methodologies, have been placed alongside those of the project as shown in **Table 2**. These are then briefly expanded below.

Table 2: Project Traits alongside FDD and Agile Benefits

<i>Project Traits</i>	<i>Project Impact</i>	<i>Benefits of FDD and Agile</i>
Busy Student	Less time to design, build and evaluate initial prototype product	Core features implemented first, reducing the priority of complex or peripheral system features.
Busy Firm	Limited opportunity for site visits and communication with the systems end user groups	No benefit offered. Developer must improvise as necessary, especially with product testing.
Limited Project Runtime	Less time to design, build and evaluate initial prototype product.	Core or primary features produced before those that are more time consuming.
Complex Problem	Difficult to build a comprehensive system in the project timeframe.	The more straightforward or foundational features of the system can be built before those that are more complex. Working software is preferred to extensive documentation, allowing fast development.
Limited Financial Resources	Systems and resources used must be financially reasonable.	No benefit offered. Developer must improvise as necessary.
Developer's Knowledge	Difficult to build a comprehensive system in the project timescale.	Core or primary features produced that reflect the developer's skillset should be implemented before those that push the developer's current understanding.

Given the small-scale development team and busy work schedules of both the developer and the client, there are several adaptations that must be made to the FDD model for it to suit the needs of this project. The fundamental difference between this project and one on an industrial level is that of scale. FDD is said to work well with a number of highly skilled object model designers and developers [149],[150]. To reflect the significantly reduced capacity of a single developer, the speed of design and development iterations throughout this project must be short to avoid unnecessary delay, particularly in appraisal, so as to produce a prototype that includes all fundamental features identified.

The need for continual communication, a core value of this customer-centric framework, is likely to prove problematic [149]. The busy academic schedule of the developer and the operational constraints experienced by the firm have already proved somewhat challenging. This is a key consideration when attempting to gratify each feature at the end of each development iteration. It is anticipated that adjustments to the continuous contact will be made, such as going to the firm's contact after the core features have been fully developed and a first prototype has been constructed - essentially developing iteratively before presenting a solution to the customer; a key consideration during the integral task of feature reprioritisation [151]. Feedback on the core functionality may be gathered before features in other categories of the MoSCoW, or features in the features-list, can be considered. Another pressure on this project is time. The benefits of FDD for this project can be seen in **Figure 15** whereby development iterations are continually reviewed against project timescales, helping to enforce project timescales and prioritise feature development.

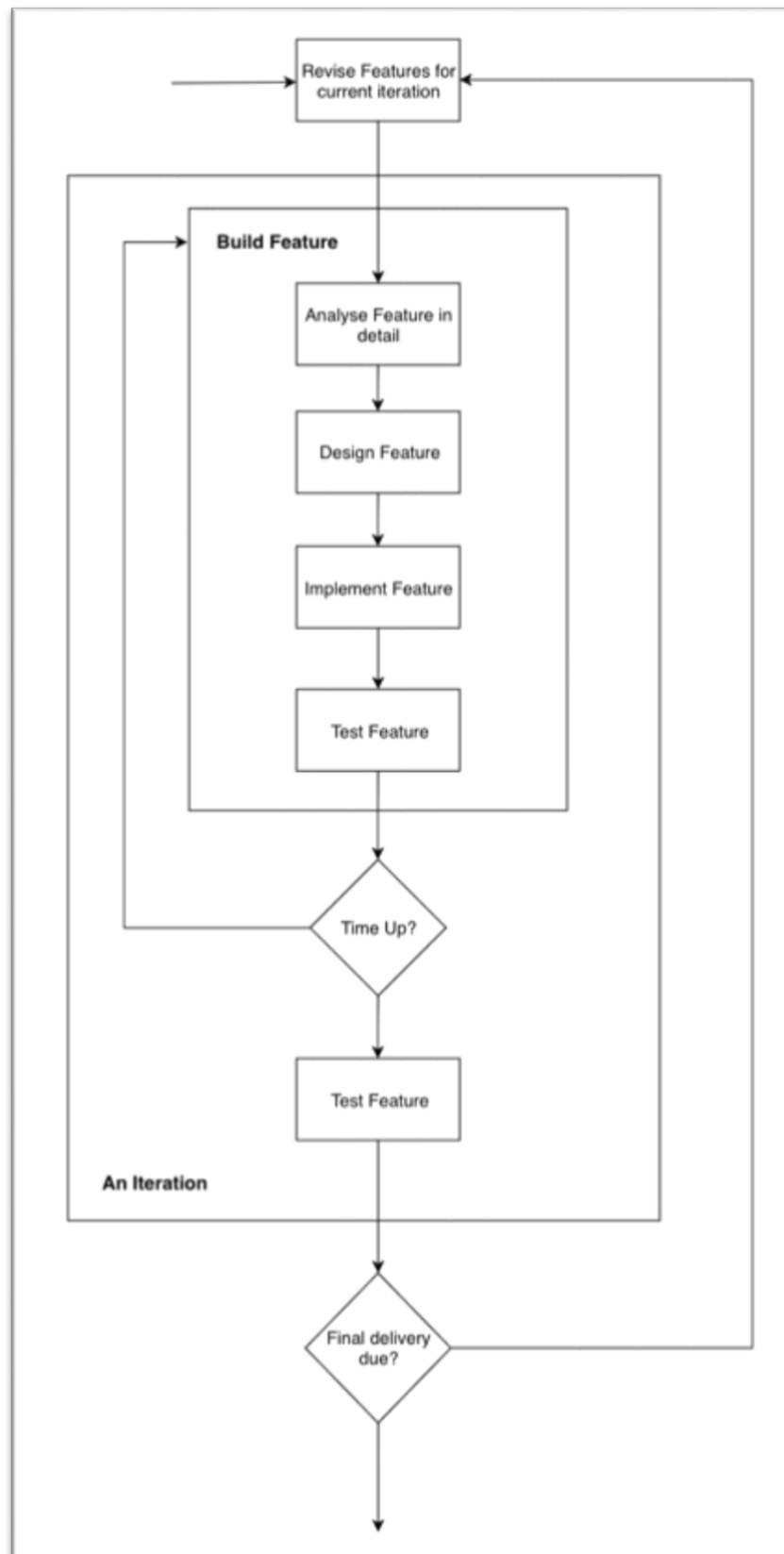


Figure 15: Time and the need to maximize iterations

3.5 Development Tools and Environments

Throughout the project a range of different software platforms have been used to aid product design and development. Sections **3.5.1** to **3.5.4** below discuss how each software tool has been selected for the project and how their use is advantageous for the developer.

3.5.1 Microsoft Access

The need to design a rigid and dependable database structure has demanded the use of a database tool. Despite not intending to use Microsoft's relational database tool, this platform allowed the developer to quickly and easily manipulate the client's data so as to create a workable database schema. The use of a GUI for this process has helped to speed up its development and quickly and easily explore possible relationships.

3.5.2 Balsamiq Interface Mock-ups

Before an interface for the application can be built, similar to that of the database schema, it must be designed. Whilst paper-based prototyping is effective, and recognised accordingly, a digital 'mock up' platform was preferred. The Balsamiq Mockup tool allows a fully functional 'playable' interface to be built up in a light and dynamic fashion. By creating a full-colour dynamic design early in the project lifecycle, feedback obtained from the client can be more precise and so has greater value than rough or sometimes unrefined wireframe sketches.

3.5.3 Microsoft Azure and SQL Management Studio

Having selected a database schema, the SQL coded database can now be built. For this, SQL Management studio on Windows, alongside Azure Data Studio (for both Mac OSX and Windows) have been used. This is connected to an instance on a Microsoft-hosted server on which the database is running. Both environments enable the construction of a database written in SQL which can then be connected to the front-end development environment or tool as necessary.

3.5.4 Appenate

To develop a useful mobile-based application in the defined timeframe, a similarly lightweight cloud-based tool will be used. There are several such options and the developer has opted to select the tool 'Appenate'. This nimble tool, allowing connectivity to an existing data source, offers the ability to capture data from registered application users to add, update and delete data records stored in

containers found connected data sources, such as SQL Server database tables selected for this project. This data can be linked to forms that support the capturing and manipulation of said data, all of which can be supervised from a single application administrator.

Given these features, this tool was selected owing to the relative similarities that can be drawn to the customer's existing software usage. GWR staff are regularly creating and updating existing form-based information, tasks monitored by managers who oversee the processing of all record-based information.

4. SYSTEM DESIGN AND DEVELOPMENT

4.1 Fulfilling the Just-in-Time Paradigm

Before the design and development of the proposed system can begin it is important for the developer to make distinct links between the theoretic JIT background explored throughout **section 2.** and the prototype system. To determine how the prototype application is supported by the theories of the JIT a table has been produced that clearly documents the relationship between the framework, the system by means of highlighting the intermediate firm practices.

Table 3: How the proposed system realizes attribute of the JIT model

JIT Theory	Current case study process	Trait of proposed system
Reduce unnecessary waste (time, resources and capacity)	An array of flat-file database and planning sheets ensure sufficient capacity on-site, that jobs can be suitably allocated and that resources are available on-site, or in transit, that can fulfil the work job.	The application is intended to offer increased efficiency the following areas: 1. Capacity management and work scheduling 2. Resource allocation and the streamlining of material warehousing 3. Offer the beginnings of a platform with which to manage and streamline work times.
A continuous improvement mindset	The firm are always looking for new ways to increase the lifespan of components and reduce the time that maintenance work is carried out. New tablets have been ordered to seeks financial improvements across the various maintenance divisions	The application offers a perspective into how numerous current processes could be digitalised so as reduce financial and environmental impacts.
Kanban system as a means to implement lean practices	Order scheduling activities are based on forecasted material usage, but the firm still relies heavily on obtaining components from suppliers at short notice.	Offers a lightweight tool that can be used to provide a paperless alternative to existing component management practices.
Poka-Yoke tools to “avoid inadvertent errors” [34],[35]	The firm use condition-based vending machines to ensure tools available to perform jobs are suitable for the work. Whilst this ensures the tools are fit for their job this is an efficiency tool and method to reduce wasted financial resources on tools that are unfit for engineers.	The application will offer a reference tool that provides structured work sign-off and job details to management in an ordered, auditable and manner.

'5S' method for greater workspace efficiency.	Strictly enforce numerous cleanliness and efficiency regulations in the engineering environment to ensure that The stores are precisely managed to ensure mandatory pieces kept at any instance are accessible, but not excessive.	The software will provide a means with which to ensure components are managed on-site in an organised and structured manner as components are intended to be assigned to an individual work job.
---	--	--

As can be seen from *Table 3*, the application aims to increase the efficiency of component ordering by means a portable, durable and strong application that can be used by a range of site staff across the physical workspace. However, the proposed application also aims to increase the efficiency of maintenance work performed at the site so as to realise benefits for the firm that can be clearly compared to those at the core of JIT. Furthermore, the company already practise JIT in the ordering of components, making the proposed system nothing new but rather an natural evolution existing practices.

4.2 Site Visit Outcomes

Having already discussed the techniques used in the requirements elicitation process in **section 3.2.2** this section presents the results of this activity. *Figure 16* shows a scanned copy of the outcomes of a team meeting held by site staff and the list of features that they might like to see in an application used to manage the maintenance processes practised on site. A formal write-up of this, based largely on discussions held with Chris Lunn after the office meeting, is outlined below and is accompanied by relevant comments from the developer as to how each feature might look in a prototype mobile app.

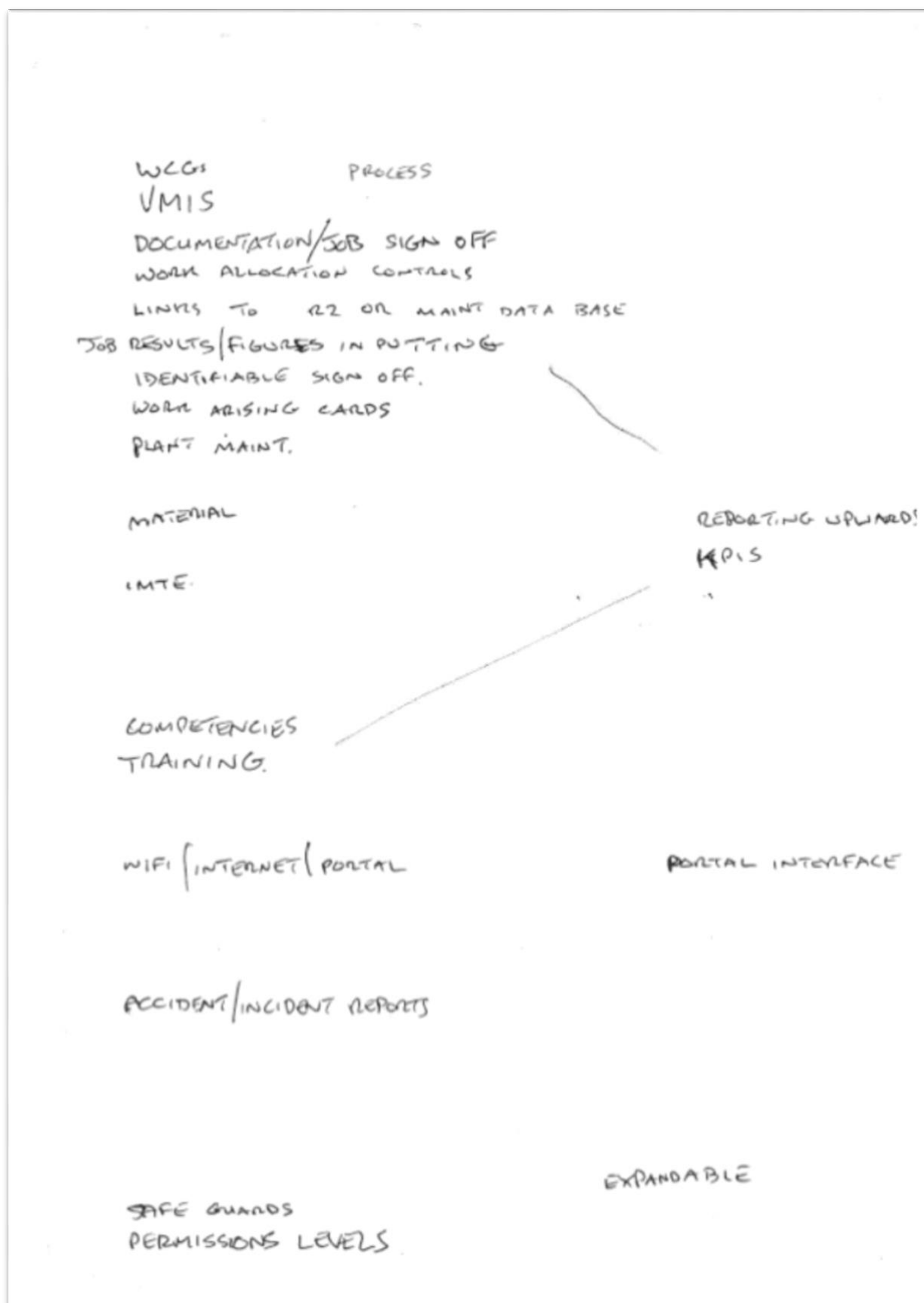


Figure 16: Application functionality list developed from a depot staff team meeting

Table 4: Developer write-up of brainstorm sheet

1. **Work Content Guide (WCG):** Large paper-based guides on performing vehicle exams that are available for the maintenance team. *The estimated costs to maintain and update at least yearly are seen in **Appendix 10.3.5**.*
2. **Vehicle Maintenance Instructions (VMI):** Other paper-based guides used advise staff performing work.
3. **Job sign off** - based on role and competency safeguards. Safety is crucial in this business. E.g. person wants to view a job, but competency level means a message appears prompting him to visit team leader. They can only sign off tasks available to them.
4. **Time as a field for exam costing totals.** Core KPI and an analysis tool that is relied upon to evaluate performance.
5. **Plant maintenance system for the buildings and equipment** on the depot. Timeline based on fixed schedules, making this much easier to manage maintenance of the facilities estate machinery.
6. **Check catalogue option** to notify if a component was required from the store. If yes, then add a time and part. If no materials required, then notes box as to how the issue was solved without parts. Also keep time field.
7. **Instrumentation Measuring Test Equipment (IMTE).** UCAS calibration laboratory needs to calibrate tools for certain jobs. Needs to be done before a certain date. Another safeguard to reduce costs of defective equipment and ensure work is done with perfectly accurately calibrated tools fit for their job.
8. **Link to GWR intranet** and internal IT systems containing base table information and a range of on-site digital resources. Important to ensure the system is secure and only accessible internally.
9. **Permission levels**, as above, for various tasks. Being able to sign off jobs, add comments and tick work in progress as complete dependant on the staff member carrying out the work.
10. **Reports integration** and viewing KPIs to check performance and view job statistics.
11. **Financial breakdown** of component totals
12. **Email Integration** and notifications. E.g. when completing a job or placing a new order with suppliers.
13. **Report an Issue.** Material non-availability, issues with a supplied piece.
14. **Annual Material Usage (AMU)** levels offer a degree of product forecasting.
15. **Show Minimum Stock Levels (MSL)** and component baseline stock requirements.
16. **Portal links.** R2 database, TRUST rolling stock allocation and status system. IT team on-site tend to avoid working with external systems. Would need to be internally managed by staff present on site.
17. **Condition-based monitoring is difficult.** If a vehicle is stopped on site for a number of days its mileage and condition is unlikely to deteriorate – meaning a predictive based system is hard to implement. Another firm at the site schedules wheel profiling based on the number of *actual* miles accumulated. Money influences the viability of preventative systems.

4.2.1 Functional Requirements

Having undertaken suitable requirement gathering activities the collected results should be evaluated and processed, after which they can be placed against a software requirement prioritisation (SRP) framework, such as MosCow [153],[154]. This structure places each requirement against its appropriate level of system importance so that it is considered either:

- **Must Have:** these features are “non-negotiable” [154]. Failure to implement these features would typically result in “failure of the entire project” [154].
- **Should Have:** These features would be nice to have “if at all possible” [154].
- **Could Have:** Features that would also be good to include but can be described as “slightly less advantageous than the ‘S’” [154].
- **Won’t Have This Time:** Features might not fall into this category because they have no obvious worth, but that because they “won’t be implemented in the current project” [154]. Despite this reduced precedence they should still be left open to possible future implementation.

Table 5: Functional Requirements List

Reqmt Facet:	Usability	Performance	Features	Processing
MUST include and/or support the following requirements:				
1.	Show a 'live' maintenance schedule that contains the following vehicles requiring maintenance at the facility: a) Power cars b) Trailer vehicles			
2.	Show a 'live' view of components ordered for the maintenance jobs a) The order(s) created to perform job(s) b) The individual component(s) that comprise job order(s) c) Append additional comments to a job d) Record the time taken to complete a job			
3.	Show a 'live' stores view that includes the following features: a) Show components available to order for planned maintenance jobs b) Configure shipping details and/or preferences for a job or stores order			
4.	The mobile device application must be designed for a tablet device and run suitably for iOS or Andriod OS platforms.			
SHOULD include and/or support the following requirements:				
1.	Ensure jobs are formally completed, and vehicle turnover maintained to ensure depot capacity, permission levels are important a) Signing off a job as an engineer b) Signing off a job as a manager or team leader			

2.	Support the following additional component management tasks: a) A list of additional components used for a job that were not initially ordered for the given job to identify “Work Arising” processes b) Show the current “strategic spare” components available in the stores for maintenance jobs as-required c) Flag defective in-bound components
3.	Provide suitable connectivity to GWR’s intranet and internal IT systems containing base table information and various on-site digital resources including, but not limited to: a) Stores catalogue spreadsheet b) R2 vehicle database (a downloaded spreadsheet containing only elements of the larger system to which initial access will not be provided) c) Planned Maintenance Schedule d) Corrective Maintenance Schedule

COULD include and/or support the following requirements:

1.	View various Key Performance Indicators (KPIs) to scrutinise the following and export to dashboard view a) The most critical components for the firm b) Average job times c) Average lead times d) Average component lifespans
----	---

WILL NOT include and/or support the following requirements:

1.	Provide a digital platform for various “Work Content Guides” (WCGs) for staff performing maintenance jobs
2.	Provide a digital platform for various “Vehicle Maintenance Instructions” (VMIs) for performing maintenance jobs
3.	Act as a maintenance schedule for on-site hardware; such as chassis jacks
4.	Act as a suitable schedule for the calibration of on-site equipment via “Instrumentation Measuring Test Equipment” (IMTE)
5.	Integrate with Network Rail ‘live’ systems via GWR’s intranet including, but not limited to a) TOPS full vehicle status and maintenance database b) TRUST vehicle allocation system

4.2.2 Non-Functional Requirements

Table 6: Non-functional Requirements List

Reqmt Facet:	Usability	Performance	Features	Processing
--------------	-----------	-------------	----------	------------

MUST include and/or support the following requirements:

1.	Application must support a very large volume data, both in repository or definition tables and maintenance records
2.	The system must be user friendly and accessible for all staff to use
3.	The application must be secure, easy to store, administer and run from the firm’s intranet.
4.	Application must provide continual availability and recoverability

SHOULD include and/or support the following requirements:

1.	The system should be easy to maintain by those who will be using it.
----	--

2.	System should enforce transaction management rules
COULD include and/or support the following requirements:	
1.	Application interface could be visually attractive
WILL NOT include and/or support the following requirements:	
<i>Not applicable</i>	

4.3 Interface Design

4.3.1 Developing User Personas

User Personas are designed to ‘simulate’ various target user groups with the aim of precisely understanding both what a particular user wishes to fulfil as well as how they intend to satisfy this need [155]. Based on a combination of demographic, biographical and *“imaginary information”* [156], this close inspection importantly offers a glimpse of their competency in the given setting as well as their *“experience levels...with computers in general”* [155]. A clear understanding of the user’s experience, thought process and system goals allows a designer to continually re-evaluate a new system by asking questions such as [155]:

“Would this fictional person really do X? What would they do instead?” [155]

“What frustrates them? What satisfies them?” [156]

Designed to *“make [our] assumptions about the target audience more explicit”* [157], a range of system users should be identified in order to help build a product that meets the often varied needs of the client [158]. The following user personas suitably reflect various types of system user, their common tasks and what they wish to get out of the system. The user research and brainstorming processes, ascertained from site visits discussed in **section 3.2.2**, has identified the following core application users:

- Various managerial roles including team leaders, managers (stores, fleet etc.) and the overall depot manager
- Engineers and technical specialists including regular engineers, lead engineers and specialist support staff
- Support staff including, but not limited to, stores staff, site administrators and other GWR personnel who are not permanently based on-site but for whom access could be necessary.

i. *Site Engineer*

Describe the User	
Personal Details	Name: <i>Mark Allen</i> Age: <i>34</i> Gender: <i>M</i> Role at GWR: <i>Regular HST Vehicle Engineer</i> Experience in Industry: <i>7 Years</i>
Professional Record	Mark graduated from the University of Bournemouth 10 years ago with a 2:1 in engineering prior to taking up his role with GWR's engineering team. Having worked for several years after graduation outside of the rail industry, Mark moved into his current role and is looking to progress further within the on-site team to an engineering Team Leader role.
Technical Skills and Preferences	Mark's broad engineering knowledge extends to computer skills and he has a moderate to highly capable level of expertise. He owns a personal smartphone as well as a work phone, the latter is used regularly to check work emails on the go and contact other members of staff around the site. Mark also has access to a desktop workstation that he uses to view email attachments containing work schedules and day-to-day staff notices
User Motivations	
Goals	Mark has several key goals that he wants to achieve when using digital systems in his role: • View work schedules • Avoid having to manage the vast quantities of documents contained in email attachments
Discomforts and Irritations	Mark has several discomforts with existing systems: • Does not like completing paperwork so wants to reduce this task as much as possible • Managing numerous versions of the same document; some of which are sent out and replaced daily.

ii. *Stores Team*

Describe the User	
Personal Details	Name: <i>Simon Fallows</i> Age: <i>42</i> Gender: <i>M</i> Role at GWR: <i>Senior Site Stores Manager</i> Year(s) in Industry: <i>16 Years</i>
Professional Record	Simon has been working in the industry for the last 16 years but has only been in the current role with GWR for the last three. Simon moved across to the stores team from the responsibility of shift Team Leader and now manages the intake, storage and output of engineering pieces. He spends most of his time on site.
Technical Skills and Preferences	Like all employees Simon has a work and personal phone. The work phone is used primarily for emails and phone calls between other members of staff when they do not answer their office phone. Simon uses stock intake sheets and component usage records, as well as material request forms (MRF) to order in fresh supplies and approve materials for jobs. To manage the ordering process a desktop machine is used, not the work mobile device.
User Motivations	
Goals	Simon has several key goals that he wants to achieve when using digital systems in his role: • Clearly see component request forms • Grant materials to jobs to fulfil request forms • View and filter component catalogue • Communicate with component manufacturers and suppliers
Discomforts and Irritations	Simon has several discomforts with existing systems: • Short notice requests (requiring high importance fast delivery) • A large site stores inventory • Left over or unused components (left over pieces from component kits) • Supply chain issues that occur out of the firm's control that directly influence delivery timetables

--	--

iii. *Team Leader*

Describe the User	
Personal Details	Name: <i>Ian Little</i> Age: <i>42</i> Gender: <i>M</i> Role at GWR: <i>Shift Team Leader</i> Year(s) in Industry: <i>18 Years</i>
Professional Record	Ian moved into the industry early in his career as a member of site support staff. He has worked with the firm for a number of years and has now worked his way up through several roles to a position of team leader. This task involves managing a number of staff members over a shift period by coordinating different activities and jobs as they arise. This sees him working with various individuals and jobs.
Technical Skills and Preferences	To manage staff across the site performing various different roles, Ian relies on the work smartphone. To fill out forms and other digital paperwork he, like all staff, has access to a desktop that is used for emails, spreadsheet tables and day-to-day administration purposes.
User Motivations	
Goals	Ian has several key goals that he wants to achieve when using digital systems in his role: • View the state of all maintenance jobs and component orders in order to balance staff and jobs.
Discomforts and Irritations	Ian has several discomforts with existing systems: • Not being able to easily find the correct work schedule • Too many emails that take time to view, save or print out • Finding staff when they are spread across the large site. Ringing their mobile or office phone does not always raise an answer.

4.3.2 Creating User Stories

Narratives help to make something more memorable and relatable as they offer a view into the thought and interaction processes of individuals [159], crucially “*as a participant rather than an observer*” [159]. These stories often take the form of a scenario whereby a particular user’s thought processes, goals and understanding, as previously documented in the user personas, play out when attempting to perform a prototypical system task. Such a narrative provides a view of a user’s position throughout and how they interact with the system, offering key design considerations but, given their imprecise nature, not how they should be met [160]. In addition, the tasks can then be attached to user personas to allow the developer to build a complete picture of an interaction that real system users will take to perform real system tasks.

“As a <type of user> I want to <some goal> so that <some action>” [161],[160]

i. Site Engineer

Mark has just completed the first task of his shift (changing a power cars traction motor) and must now complete the job’s accompanying paperwork back up in the site office. To do this Mark heads back up to the office and out of the workshop. Back in the office he unlocks his desktop computer and opens up Windows email client. Mark has a large number of emails in his inbox and uses the search box to attempt to find the correct email. He knows that it has come from the Team Leader and so searches for this sender. Having looked inside several emails, and attachments, Mark finds the word file he is looking for, downloading it to the machine in order to complete it.

Having finished the sheet Mark sends the amended digital copy back as an attachment to the original sender whilst also printing a copy to add to the paper-based records held on each vehicle. Mark then performs another search to find the daily schedule email, previewing the attachment to view the daily job list and next possible task.

ii. Stores Team

Simon has just arrived on site and signed in when he receives a call from a colleague letting him know that he has dropped Simon a MRF email. When Simon logs into his desktop machine and opens his email he can see the unread message, subsequently downloading and

opening the attached MRF file. The request is for a series of relay battery boxes for a power car that is due to arrive at the site in several days for a B Exam. Firstly, Simon wants to check the stores to see if any components for this job are already being held.

Having checked the storage location in the stores, and finding that none are currently being held, Simon must place an order with the component suppliers to arrange for a short notice courier delivery. To do this he opens up the desktop browser and signs in to UniPart Rail's portal, seeing that a similar order already exists. Rather than creating a new order Simon edits an existing delivery to add the extra component packages. Having made this adjustment Simon then prints the amended invoice and alters the existing order record in the firm's digital order spreadsheet, contained within an Excel database file. He also forwards this new invoice to management to flag the order change. Finally, he calls the member of staff back to notify them of the component and order status and that the supplier expects the component to have arrived in the stores before the vehicle exam is due to commence.

iii. Team Leader

Ian has just been down the maintenance shed to speak to a colleague about a problem that he has overcome and has just sat back down at his computer to check for any new emails. He sees that an attached sheet from the vehicle maintenance manager has arrived regarding the planned vehicle schedule that they have had sent down from the head fleet planners off-site. Ian prints off a copy to hand around staff in the office before copying each job from the list and adding it into the depot's operational schedule for each day. He ensures that no conflicts of vehicle, capacity or staff prevent each piece of work being done and contacts staff at other depots when schedule conflicts arise. As such, some of the jobs on this list are added, but subsequently removed to ensure a workable schedule. Any work required that is not manageable is recorded and sent back to the depot manager. Having completed this mapping process Ian can save the excel schedule documents, sending them out to all staff by email and preparing printed copies for the office notice board.

4.3.3 Wireframe Designs and Design Patterns

i. Navigation

c) Two-pane selector

This design pattern places a selection list, typically a menu structure, on one side of the screen alongside a larger pane that displays the contents of the selected menu item, a pattern most clearly seen in Apple's iOS settings application on their iPad devices, as illustrated in

. Given the size of the panes this configuration is suitable for larger screens and so is rarely used on smartphones [155].

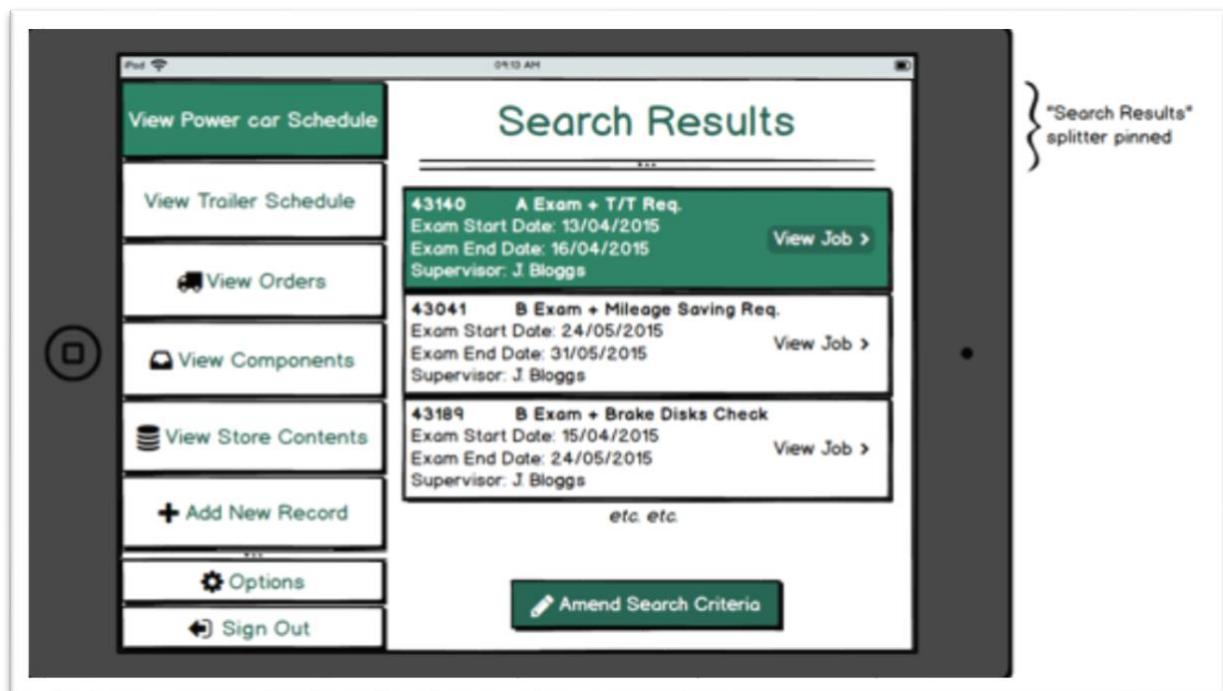


Figure 17: The two-pane selector pattern for primary menu selection

The two-pane configuration offers a reduced memory burden as a user needs just a glance at the left panel to see their current location, acting as a 'You are here' reference [155]. A two-panel approach is appropriate when the primary menu consists of *"objects, categories, or even actions"* [155], all of which are present in the wireframe designs. This concept is a minor adaptation of the springboard pattern, both of which are often referred to as the launchpad due to typical use in a primary menu from which users jump into the main application [162].

This approach is suitable because such a pattern, thanks partly to the nature of the larger display, *"allows content to be front and centre"* [69] - particularly useful due to the record-based nature of this application. It is important to note however, that the secondary pane should always *"correspond to the clearly identifiable selection in the primary screen"* [163] so that users *"can clearly understand the relationships between the panes"* [163]. This factor is clearly present in the prototype system designs.

d) List Pattern

The left-hand pane lists items continually, presenting them as a single interaction element and so are best suited for *“presenting a homogenous data type”* [164], a control referred to as the ‘Navigation Bar’ on iOS [69]. This control, used here to fulfil the role of a left pane, supports the ‘F-scan’ user convention whereby users scan left-to-right so as to take in each list item, and downwards to view all the options contained within [164]. Such a common control here works when it is short, supporting quick user comprehension and navigation within the app - but it is worth noting that unnecessarily lengthy lists can result in a diminishing user attention span [164].

ii. Capturing Form Input

The basis of the application is the need to create, manage and fulfil time-based events for both a schedule and order system, thus a straightforward date-style picker is used, as illustrated in **Figure 18**. This control is easy to understand and use, and is clear due to its ‘popover’ appearance. A picker is suitable as a date selection control because of the larger screen size and the clarity that a larger picker ‘wheel’ can provide. It is also situated *“in proximity to the field being edited”* [165].

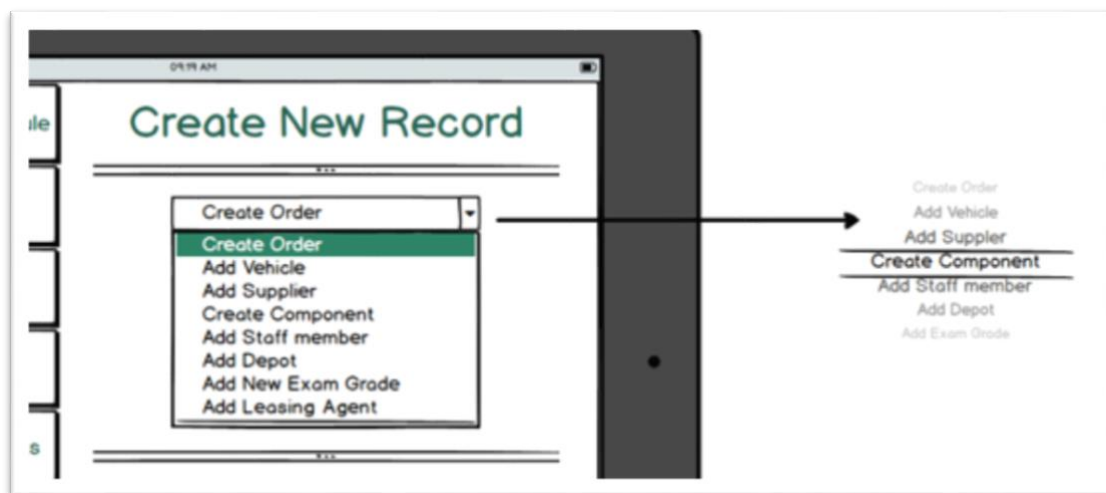


Figure 18: A possible use of a picker-type spinner

In numerous locations throughout the interface a simple switch is required, a perfectly suitable ‘on’ or ‘off’ control that can be linked to fields in the underlying database of the ‘BIT’ SQL type, especially given they are *“intended for use in tables, such as in a list of settings”* [166].

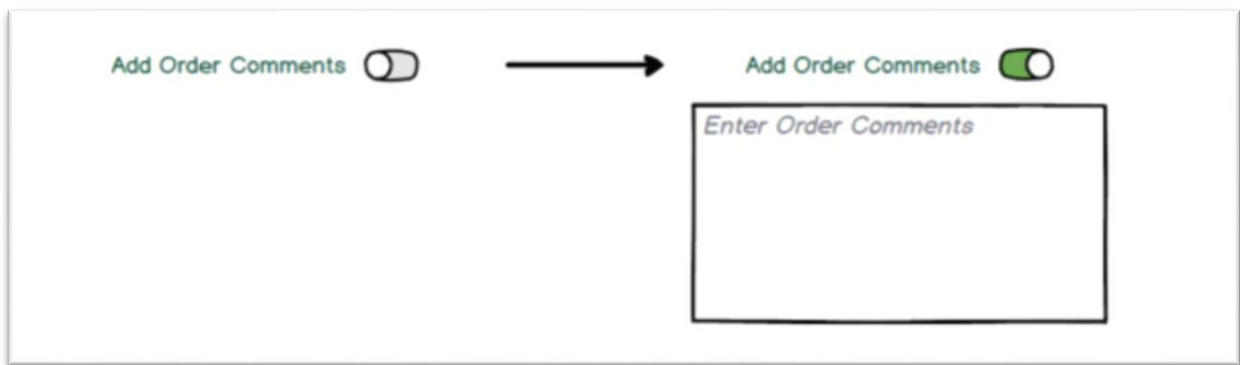


Figure 19: An iOS-style switch has been used

Menu items are used throughout the app, in order to perform a refined search for existing job entries or to add a component to an existing or new order. Such options, known as tables, frequently take the form of sections or groups of items presented as a “*scrolling, single column list of rows that can be divided*” [167] in which various controls can be placed. **Figure 20** shows how this control could be used for searching component name [168].

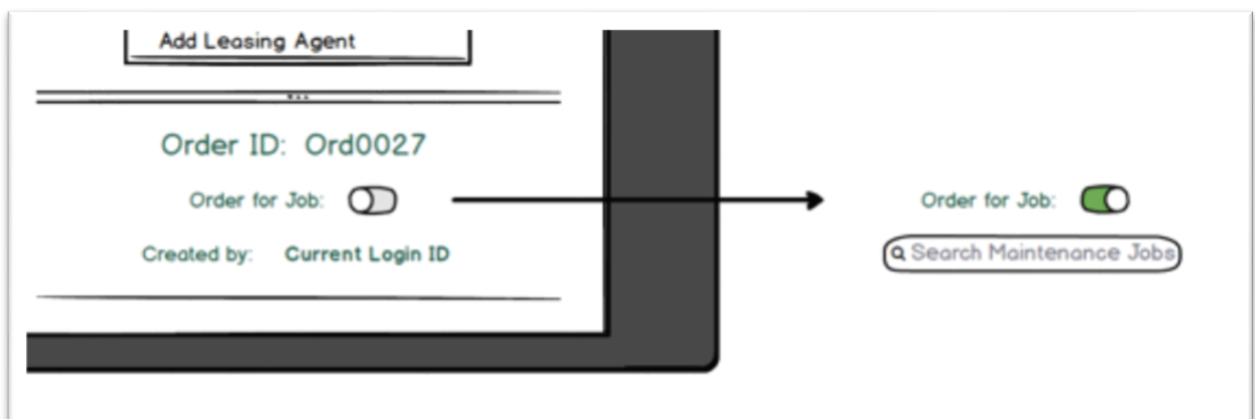


Figure 20: Searchable lists for component names or IDs, similar to ‘Contacts’

The option to search by component name or ID could be further refined by using a scope bar [168] as illustrated in **Figure 21**.

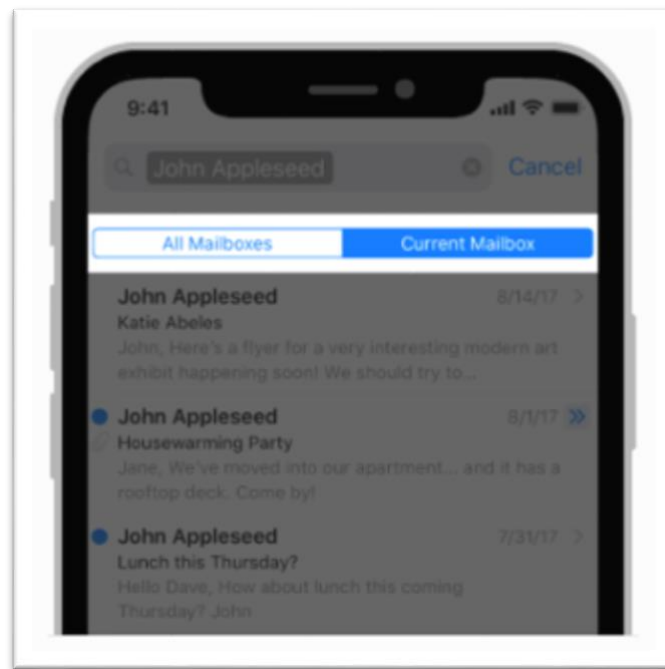


Figure 21: Search bar with scope 'toggle' bar [168]

A similar grouped table showing customisable widgets has been used in the initial application designs to enable users to manage order contents. Once components have been added to an order, they can be reorganised by dragging the hamburger-style 'handle' or removed by tapping the remove icon, as shown in **Figure 22**. This customisable menu list has been used in conjunction with a search bar to allow a member of staff to quickly find items and add them to a controllable list.

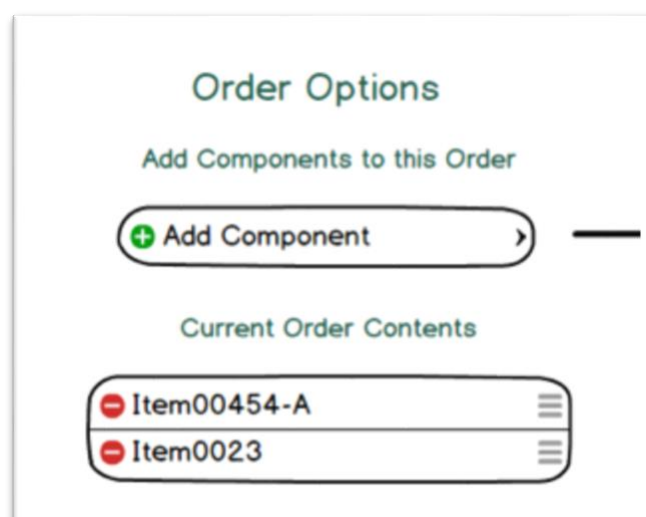


Figure 22: Customisable list with 'add', 'remove' and 'reorder' in iOS 'Control Centre' settings.

4.4 Application Development

4.4.1 Develop Overall Model

A software solution developed in a Feature-Driven manner, as previously discussed, can be started by building up various views of system functionality so as to create a broad high-level system model. A Unified Modelling Language (UML) diagram [169] for this project has been developed and is illustrated in **Figure 23**.

The diagram archetype (also known as its stereotype) is that of system classes, a visual representation that is further enhanced by applying Coad's UML colour categorisation [170]. This annotation method categorises each class (or archetype) into a particular role within the system using `<<system role>>`, a notation that is easy for both the developer and the customer to understand [171].

- Pink, or `<<moment-intervals>>` classes represent *"events or activities"* [171] within the system that must be enforced so as to provide business goals, typically scheduling, booking and other actions that can be described as event or time specific events [171]. The maintenance schedule is a notable example of this role in this system.
- Those coloured yellow and denoted as `<<role>>` represent classes that *"participate in an event or activity"* [171]. In this system a leasing agent firm who leases vehicles to GWR (and who advocate the use of certain components) is an example of a such a relationship as they are interact with a green or `<<thing>>` [171].
- A green class covers places, people and physical locations. Being so broad such classes are noted as green and then tagged as `<<party>>`, `<<place>>`, or `<<thing>>`. All green classes noted in this UML view represent physical "things" such as a component or a vehicle.
- A blue class represents a `<<description>>` and as such typically contains class methods. Owing to the purpose of object methods, *"a green (for thing) followed by blue (for description)"* [171] is customary. This can be seen in this scenario where Grade plays the role of a descriptive class for Component.

The action of colouring classes helps the developer to understand how classes of a certain role interact. Such roles can provide a greater understanding of how the class will actually be implemented, because *"the four colours correspond to its characteristics"* [170] and its attributes, methods and interactions expected in objects of its type.

Multiplicity notes the number of each class instance that can be present in a given class association [171]. These vary throughout the prototype system so that a single component supplier can supply 0,

one or many (“*”) order deliveries, whilst a single depot is likely to have many employees, several team leaders but only one site manager. These two examples represent composition aggregation whereby “the whole and parts have coincidental lifetimes” [172]. An instance of Job_Order can still exist if a supplier is dropped (in that it will remain on a system; albeit without a firm to fulfil it). The same is true for a site manager who should remain a member of staff even if their depot is removed from the system.

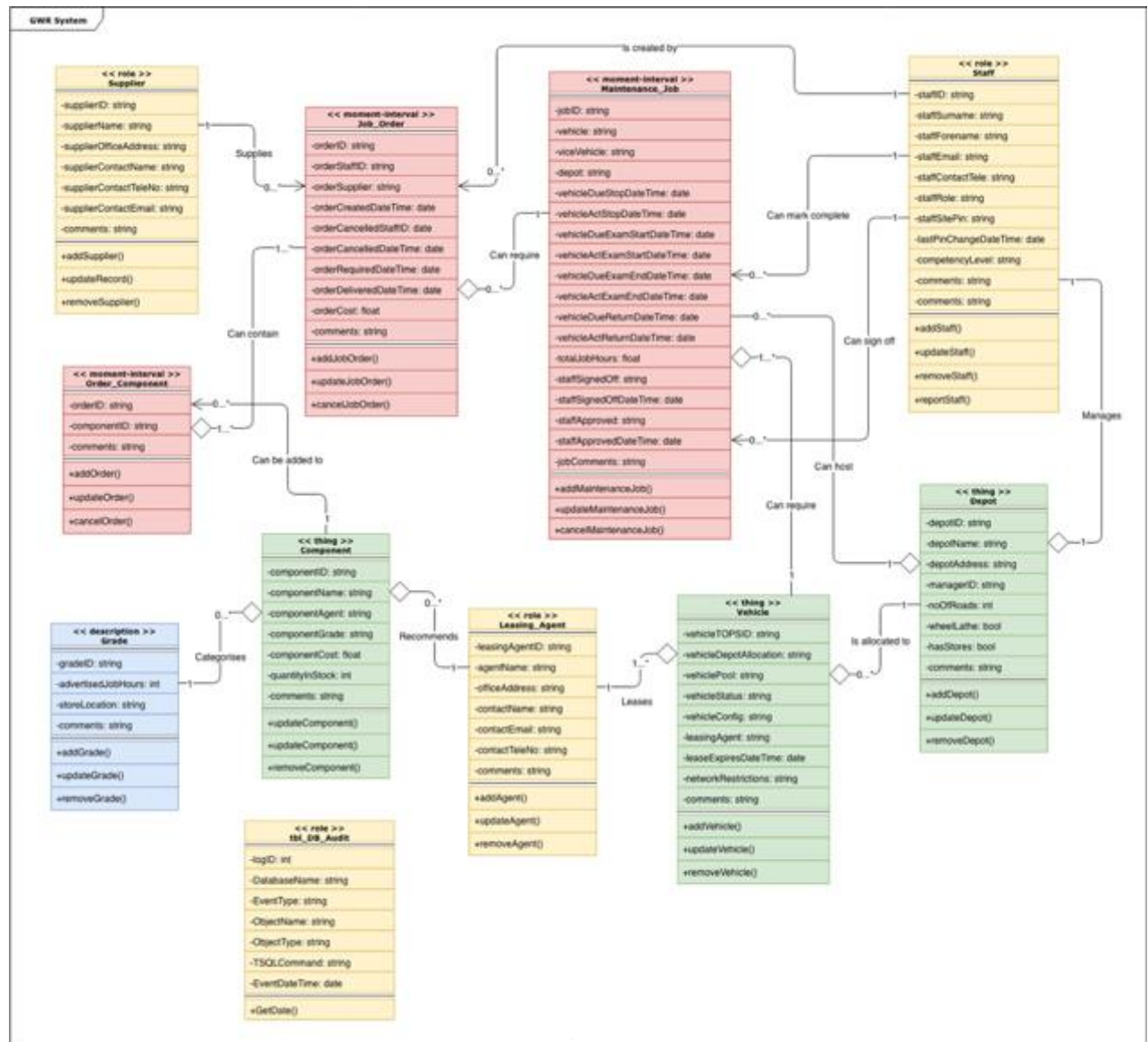


Figure 23: A possible system class UML diagram

Whilst this is a useful modelling tool, it can’t be used to understand system processes as it is too generic [169]. This is where a feature-list, the next stage of the FDD lifecycle, has real value. Alongside the architectural modelling processes, initial UI mock-ups can be developed [173]. Though these would traditionally be mono-chrome wireframe prototypes, full colour designs have been produced and are illustrated and discussed in **section 4.3**.

4.4.2 Build Feature List

Based on the construction of an overall system model, the precise elements of the product can be defined - based partly on this model and previous requirement gathering processes [174]. Coad's original United Overseas Bank project in Singapore helped give rise to the idea of a "feature" as a core software development element [174], though it is somewhat different to a requirement as it is "*schedulable*" [134]. They were written in a form that is accessible to all project stakeholders and sponsors, and therefore are likened to a requirement; commonly following this construct:

"<action> [a|the] <result> [of|to|for|from|...] <object> [with|for|of|...] <parameters>"
[174]

An example of one such feature-list element for this project being:

"Create a maintenance job for a power car in the schedule"

Having documented these requirements, they can be grouped into feature-sets whereby "*business related features*" [175] appear as a more generic requirement description. For example, the task of adding a new job for a vehicle to the schedule should be considered a single task that comprises the broader duty of managing a day-to-day maintenance schedule.

"<action><-ing><a(n)><object>" [175]

One fundamental set for this project being;

"Manage the maintenance schedule" [175]

Each individual feature-set feature (shown in the right-hand column), and the appropriate feature-set in which it is grouped (listed down the left) is shown in **Table 7** and for this project and future development time boxes.

Table 7: Feature-list features and grouped feature-sets in this project

<i>Timebox No.</i>	<i>Feature Set</i>	<i>Feature-list feature</i>
--------------------	--------------------	-----------------------------

2 (database creation forms time box 1.)	Manage the Maintenance Schedule	Create a power car maintenance job in the schedule
		Create a trailer car maintenance job in the schedule
		Remove a power car maintenance job from the schedule
		Remove a trailer car maintenance job from the schedule
		Edit an existing power car job in the schedule
		Edit an existing trailer car job in the schedule
3	Manage Component Ordering	Add an order for an existing power car job
		Add a component to an existing order for a power car job
		Remove a component from an existing power car job order
		Remove an order for an existing power car job
4	<i>Project Iteration Bug Fixes</i>	Fix errors that have been found with the initial system prototype raised in <i>Table 7</i> .

Table 8: Feature-list features and grouped feature-sets **beyond project scope**

4	Manage the vehicle fleet	Add a new power car to the fleet (vehicle being leased)
		Remove a power car from the fleet (vehicle going off-lease)
		Update the status of a power car
		Re-allocate a power car to a different pool
		Re-allocate a power car to a different 'home' depot
		Add any operational restrictions as appropriate
5	Manage the component catalogue listing	Add a new component to the list of required pieces
		View the currently available pieces that can be taken out of the site stores
		Update the details of a component in the required pieces list
		Remove a component from the list of required pieces

Owing to strong requirements-gathering processes, and subsequently thorough brainstorming and prioritisation with the customer, the development of features and their feature-sets has been significantly advanced. The initial requirements-gathering process and the requirement lists developed during this process has been previously discussed in **section 3.2.2**.

4.4.3 Plan by Feature

Having developed a concise set of features, a more precise development schedule and taskforce that can fulfil the various elements of system development can now be constructed that can meet the challenges of producing each feature-set.

This task takes the feature sets and assigns these functionality groups to small development teams believed to be the most suited to implementing the precise feature-set features contained within. However, for a single developer setup the allocation of resources and size of timeboxes should be appropriate to the workload capacity of the individual to ensure that development is realistically attainable.

This process often includes outward perspectives in terms of *“meeting any external [visible] milestones”* [148] and any tasks involved with releasing system prototypes to such stakeholders.

4.4.4 Iterative Development: Timeboxes

Each timebox has its own set date range and requirements so as to assign work to a specific timespan [134] and concentrate on manageable chunks of functionality. This distinct time period contains the contents developed within, appropriate system designs for the proposed functionality, key points during the practical implementation and, in the case of the database model, subsequent testing procedures. The following timeboxes are organised throughout **section 4.4.4** as follows:

- Timebox 1: Database development and data governance
- Timebox 2: Creating, managing and searching the job schedule
- Timebox 3: Creating, viewing and searching component orders and their contents

i. *Timebox 1: Database Development*

Start Date: 25 February 2019

Anticipated End Date: 15 March 2019

Actual End Date: 31 March 2019

The process of documenting the development of the maintenance and order scheduling database primarily covers the development of its constituent tables, data types and information storage and how the relationships have been constructed in a relational database format. Such databases are commonly found in ERP software, as previously highlighted in a way that *“internal [and external] users are given a variety of Web-portal interfaces to interact with the back-end databases”* [2].

a) Contents

- Create database tables
- Create and insert database records
- Create and define strong database relationships to provide a robust relational structure
- Implement an initial data governance mechanism
- Create a successful connection between the Azure and Appenete servers

b) Feature Design and Implementation

A use case diagram, unlike the overall object model, provides a deeper understanding of a user’s interaction with the system, and a goal that they wish to achieve. This short, descriptive paragraph is relatable and has value for that user in such a way that *“developers ... have a broad sense of what lurks inside”* [169].

A common way of providing this understanding in the context of a database is to build an Entity Relationship (ER) diagram that shows the database tables, fields and their data type and size. These diagrams also show table relationships and optional multiplicity.

The ERD diagram illustrated in **Figure 24** shows how the developer has played around with the concepts of an ordering-based system. **Figure 25**, also below, shows the final version. The second schema show how Suppliers, Components, Staff and Depots facilitate the creation of a work schedule and how external orders are created and subsequently managed by site staff that allow vehicles to receive mechanical attention that they require.

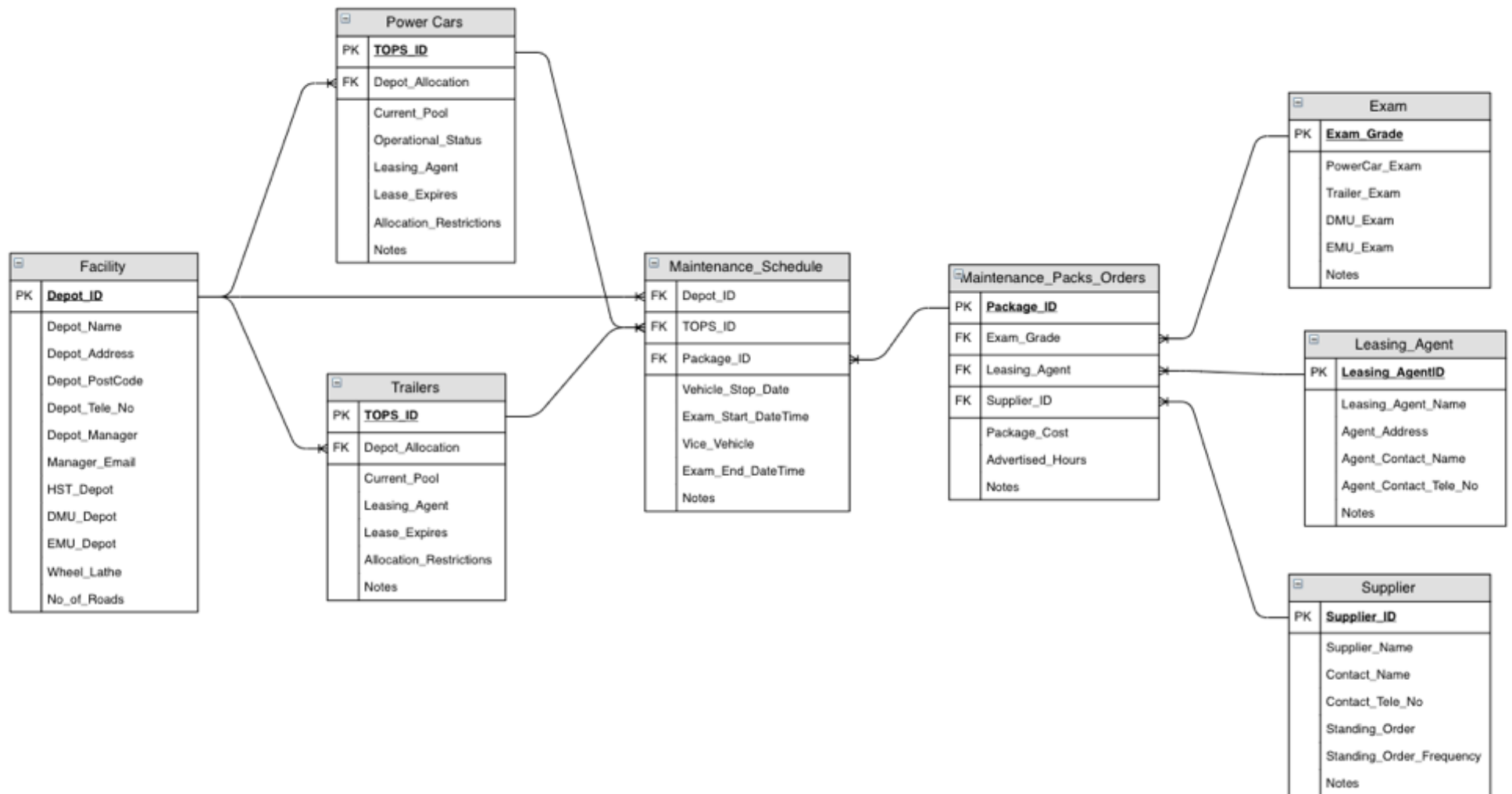


Figure 24: An initial draft database schema that was quickly dismissed

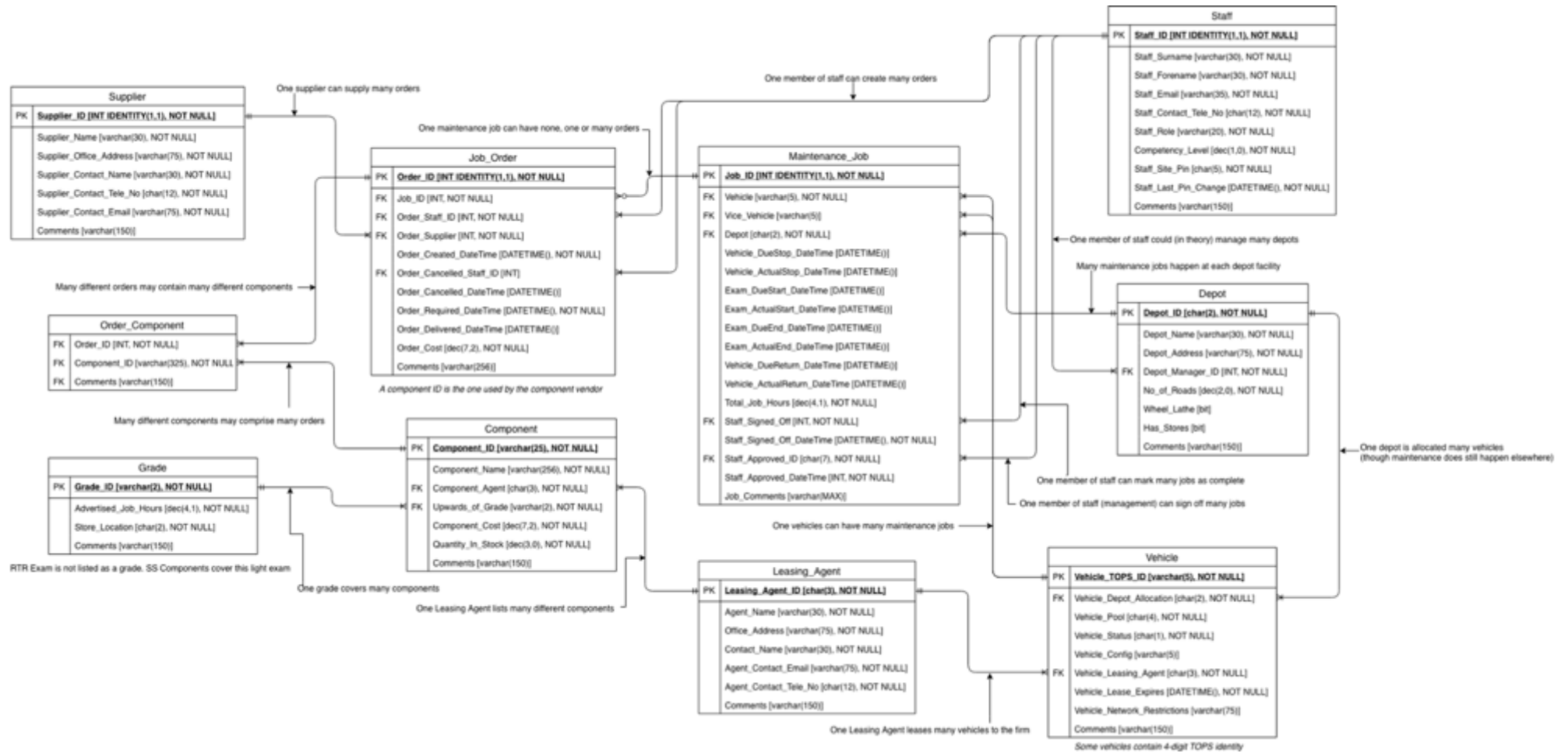


Figure 25: The final entity relationship diagram (ERD) for this scenario

Having played around with various database schemas the developer finally fixed on the schema illustrated in **Figure 25**. The next step is to replicate the structure in SQL by creating the database in the Azure cloud. Below is a brief discussion of several key tables in the database that are central to the project scenario.

Maintenance Jobs

The maintenance job table is central and records maintenance work that vehicles will and/or have received at each site. For each job entry in this schedule a new Job_ID is created that places a vehicle at a Depot, recording the following key information.

- Any vehicle, if applicable, that is being swapped with the incoming vehicle. This has been set to a **NULLABLE** field owing to this optional condition. The **CHECK** constraint ensures these values are not the same, implying that a vehicle is swapping with itself.
- A series of **DATETIME** fields capture both the planned and actual times of the vehicle's arrival at the site, the start of its maintenance work, the end of the job and its time of departure. All fields here accept **NULL** so as to accommodate unplanned work (whereby 'Due' fields are empty) or cancelled jobs (whereby 'Act' fields are not required).
- Staff are required to sign off work when complete, a process that is captured by a timestamp stored in Signed_Off_DateTime.
- A job marked complete should be accepted and approved by management. This is accepted in a text-field (Staff_Approved_Job) in which a staff member must type their ID.

Job Orders

A table is needed that creates an order instance and places it against a maintenance job, recording the appropriate information regarding its creation, adjustment, delivery or subsequent deletion information. Job_Order, referencing a Job_ID, records the member of site staff, to fulfil the order in the field Order_Supplier_ID and any cancellation information such as the member of staff who performed the action and when. A field is added that records the required date of the order and its subsequent delivery time, both of **DATETIME** that accesses the **GETDATE()** function.

Order Components

Order_Component acts as an intermediate table to split up a many-to-many relationship that is anticipated to occur between these system entities, a condition that can be seen in the following statement:

“Many components will be added to and appear in many component orders”

To satisfy database integrity rules this relationship needs to be broken. Each unique Order_ID in the table Job_Order is referenced in Order_Component where it can appear none, one or many times so as to place a Component_ID as many times as is required against the order. No composite key, an integrity rule that defines *“a unique combination of columns”* [176], has been applied to allow a member of staff to place a component in an order more than once.

Instead of using an alphanumeric identifier field in maintenance job, component order or staff tables, these tables have an auto-incrementing integer value. It is more straightforward for a user to add, query and recollect fields consisting of a single data type, such as a one, two, or three-digit number, rather than an identified comprising a mixed format, such as the ‘ORD0031’. To configure this, the transact SQL keyword of **IDENTITY()** has instead been used. Illustrated below is the revised ‘Order_Job’ table with this primary key format.

```
Order_ID [INT] IDENTITY(1,1) NOT NULL CONSTRAINT PK_Order_ID PRIMARY KEY,
```

In several instances data inserted into the tables with this configuration was in the incorrect order. Simply clearing the table using the following statement did not work as intended as any subsequent entries saw the ID field continue to increment from its last value. To reset the ID value seed before re-inserting records the following SQL line was executed.

```
DBCC CHECKIDENT ('Maintenance_Job', RESEED, 0);  
GO
```

Database Data Governance

Whilst Transact-SQL supports its own Database Auditing tool, “Server Audit”, that creates a whole system log [177], this prototype solution deploys a shadow table; a user-defined table running off a trigger [178]. This system level event trigger captures structural level data definition language (DDL) actions [2] performed on the database. This solution however doesn’t record the state of the target object before or after the SQL command that was executed on the system. Despite providing broad and widespread coverage, and providing a log of database events, it is suggested that such a process is unsuited for data versioning [179], a time-based system involving a user-defined table *“designed to keep a full history of data changes”* [180].

In this scenario a table such as this can be used to record and subsequently query the actions performed on a database by all members of staff. This mechanism helps to provide a reliable, system-

wide action log that could be used for reporting or security purposes. The SQL CREATE TABLE statement and accompanying DML trigger and INSERT code can be seen in **Appendix 10.5.1** and **10.5.3** respectively. *The SQL to create the DDL audit table was taken directly from a database audit solution offered by Worthen [178].*

Create a successful connection between the Azure and Appenate servers

To create a valid connection between the SQL database and the front-end builder Appenate, use of the databases connection string is required. This string provides information as to the name and location, administrator account, querying and encryption settings configured for the database location.

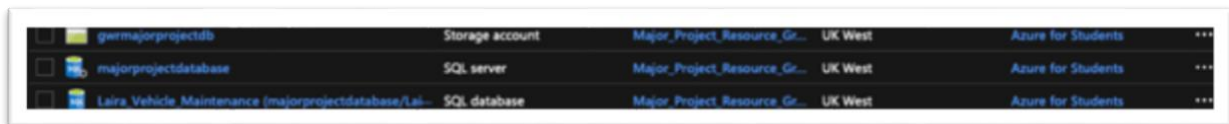


Figure 26: Azure account resources

Figure 26 shows the database on the server with Figure 27 illustrating the connection string created by Azure in which the user must add the details of the defined database administrator.

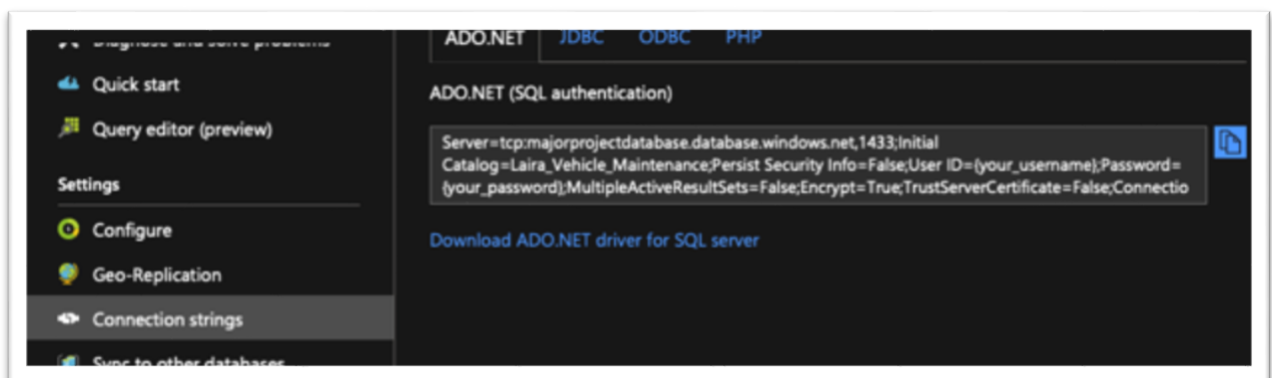


Figure 27: Database connection string to access database

The connection string can then be copied into and adjusted in the Appenate connection page to create the link between Appenate and the database container.

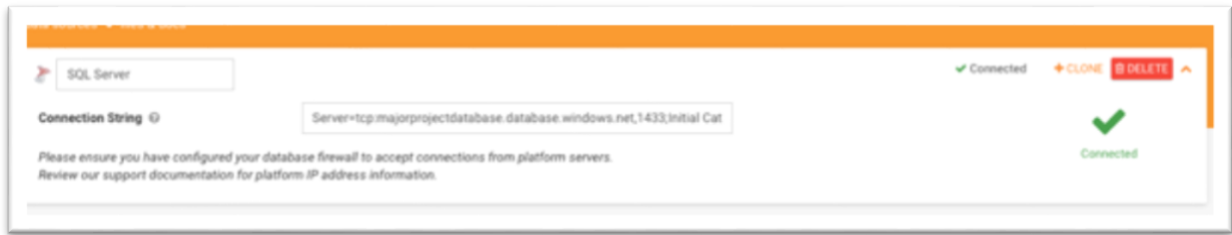


Figure 28: The SQL Server connection with the string providing a valid connection to Azure resource

Having created a valid connection, the data contained within the database must be extracted in Appenate to the 'Data Sources' page in order to use the data held within the database in a mobile application. This view with several connected tables is illustrated in **Figure 29** with the 'row' view of the table contents shown in **Figure 30**. From these views it is evident that the database has been connected and the data within the tables is accessible. The table contents can now be used in the application's various screens.

☐	Name	Type	External Id	Rows
☐	Published Docs	System	PUBLISHEDDOCS	1
☐	App: Incomplete Tasks	App	APPTASKS	1
☐	App: Entry History	App	APPHISTORY	1
☐	App: Entry Drafts	App	APPDRAFTS	1
☐	Active Users	System	ACTIVEUSERS	1
☐	Grade	Connected		10
☐	Vehicle settings • rows	Connected		44
☐	Supplier	Connected		10
☐	Staff	Connected		17

Figure 29: Connected database tables

vehicle_tops_id	vehicle_depot_allocation	vehicle_pool
0	1	2
40734	LA	EFTC
41008	PZ	EFTC
41166	PZ	EFTC
42067	LA	EFTC

Figure 30: Table source row view

c) Database Testing

Section **c)** concentrates on the testing of the database schema so as to ensure that its structure provides both the appropriate degree of rigidity, security and dependability expected of such a system and that it can enforce the various business and scenario-specific needs of the product client. To ensure its suitability the database will be tested for security by means of the audit and system log table, and then for its business role enforcement by performing precise queries on the database contents. The queries selected to be performed on the database reflect actions that GWR site staff are likely to be performing on a regularly basis.

Test 1: Database Security Mechanisms and Action Log

“Ensuring that structural or administrative changes to the database are logged for managers and senior staff to observe”

To check that the database table WholeSystemLog records any alterations made to the system a user will alter an existing table by adding a new column named Order_Accepted_DateTime to the existing table Job_Order.

Firstly, the table was updated by the SQL user login *henrycobbold* to append the new column as illustrated below:

```
ALTER TABLE Job_Order
ADD Order_Accepted_DateTime DATETIME; --nullable to accept rejected orders
```

The query was executed successfully, after which a SELECT * query returned the new column in this table, all rows of which are currently set to NULL.

Order_Cost	Comments	Order_Accepted_DateTime
575.59	NULL	NULL
1238.50	NULL	NULL
81210.99	cancelled - traction motor order...	NULL
418.00	NULL	NULL

Figure 31: A new column created on the Job_Order table.

Running the following SELECT query on 'WholeSystemLog' returned the following SQL. The single line result shows the username, time and full SQL command by which the new column was created as part of the ALTER statement. A capture of this table can be seen in *Figure 32*.

```
SELECT * FROM WholeSystemLog
WHERE EventType = 'ALTER_TABLE';
```

RESULTS								
	LogID	DatabaseName	EventType	ObjectName	ObjectType	TSQLCommand	EventDateTime	LoginName
1	10	Laira_Vehicle_Maintenance	ALTER_TABLE	Job_Order	TABLE	ALTER TABLE Job_Order ADD ...	2019-04-22 10:07:19.270	henrycobbold
MESSAGES								
11:11:44	Started executing query at Line 20 (1 row affected) Total execution time: 00:00:00.033							

Figure 32: The returns returned WHERE the action was ALTER_TABLE showing the SQL execution

However, there is currently no database structure that allows system administrators to precisely view changes to data contained within the tables over a given period of time. The process of duplicating every database table to create a mirrored DML-capturing object is advised so that a targeted field of a CRUD action can be recorded before and after code execution. However, due to the size of the database and the importance of highlighting key functionality, this process has been omitted.

Test 2: Querying Database Objects

To rigorously test the database schema a series of calculated SELECT queries have been performed to ensure that the database has been configured in a durable and robust manner.

Test 2a: Viewing all maintenance jobs for a given vehicle

It is highly probable that site staff will wish to view all the job records in the system that were carried out on a specific vehicle. The following query has been created that selects the vehicle ID, the vehicle being replaced for this piece of work, the exam or work start time, the exam or work end time, the jobs length and the member of staff who signed the work off and at what time. The query then searches by the Vehicle ID field to return records that provide a match, as shown below in the SQL view and *Figure 33*.

```
--jobs for a vehicle
SELECT Job_Vehicle_No, Vice_Vehicle_No, Exam_Act_Start_DateTime,
       Exam_Act_End_DateTime, Total_Job_Hours, Staff_Signed_Off, Signed_Off_DateTime
FROM Maintenance_Job WHERE Job_Vehicle_No = '43140';
```

RESULTS							
	Job_Vehicle_No	Vice_Vehicle_No	Exam_Act_Start_DateTime	Exam_Act_End_DateTime	Total_Job_Hours	Staff_Signed_Off	Signed_Off_DateTime
1	43140	43087	2018-03-27 02:00:00.000	2018-03-27 12:00:00.000	5.5	1	2018-03-27 19:25:00.000
2	43140	NULL	2018-04-10 19:00:00.000	2018-04-13 06:00:00.000	200.0	5	2018-04-13 07:40:00.000

Figure 33: Two jobs have been carried out on vehicle 43140

Test 2b: View all jobs completed between a certain timeframe

Another common information set is anticipated to be the number of jobs completed on site over a given timespan. To test this, a query has been developed that selects all rows in the table 'Maintenance_Job' where the 'Exam_Act_End_DateTime' field contains a DateTime that falls between the users desired date range, using the keyword BETWEEN. The results are then ordered with the smaller jobs (identified by length) displayed first.

```
--jobs completed between certain dates ordered by their length
SELECT * FROM Maintenance_Job WHERE Exam_Act_End_DateTime BETWEEN '01-Apr-2018' AND
'10-Apr-2018' ORDER BY Total_Job_Hours DESC;
```

The SQL command executed on the database is shown above with the five results displayed below in **Figure 34**.

	Exam_Act_End_DateTime	Vehicle_Due_Return_DateTime	Vehicle_Act_Return_DateTime	Total_Job_Hours	Sta
000	2018-04-06 15:00:00.000	2018-04-06 02:00:00.000	2018-04-06 17:00:00.000	6.0	4
000	2018-04-06 22:50:00.000	2018-04-07 02:00:00.000	2018-04-07 02:00:00.000	7.0	6
	2018-04-04 00:45:00.000	NULL	2018-04-04 02:00:00.000	10.0	14
000	2018-04-09 23:30:00.000	2018-04-10 05:00:00.000	2018-04-10 02:00:00.000	100.0	1
000	2018-04-05 00:30:00.000	2018-04-06 02:00:00.000	2018-04-05 10:00:00.000	270.0	9

Figure 34: All Jobs completed at the facility between April 1 and April 10

Test 2c: View jobs that were signed off more than three days after the jobs stated end date

During the requirements elicitation stage, the client identified their continual need to identify outstanding work so as to prioritise the completion of paperwork associated with the work. This enables the site to finish part-complete jobs in an efficient and timely fashion.

This query aims to identify pieces of work where the difference in time between the practical completion of the job and the assigned staff signature was greater than a certain timeframe; in this case 5 hours. The SELECT clause defines the target fields alongside an additional calculated field of the function DATEDIFF that specifies the two columns and the temporal gauge, in this query 'hour' as opposed to 'day'. In the query this field has an alias name of 'Hours_After_Signature'.

```
SELECT Job_ID, Job_Vehicle_No, Exam_Act_End_DateTime, Signed_Off_DateTime,
DATEDIFF(hour, Exam_Act_End_DateTime, Signed_Off_DateTime) AS
Hours_After_Signature
FROM Maintenance_Job WHERE DATEDIFF(hour, Exam_Act_End_DateTime, Signed_Off_DateTime)
> 5 ORDER BY Job_ID;
```

RESULTS					
	Job_ID	Job_Vehicle_No	Exam_Act_End_DateTime	Signed_Off_DateTime	Hours_After_Signature
1	2	43140	2018-03-27 12:00:00.000	2018-03-27 19:25:00.000	7
2	5	43004	2018-04-05 00:30:00.000	2018-04-05 12:00:00.000	12
3	18	43196	2018-04-16 02:00:00.000	2018-04-16 08:15:00.000	6

MESSAGES	
12:37:49	Started executing query at Line 49 (3 rows affected) Total execution time: 00:00:00.126

Figure 35: The DATEDIFF function shows three jobs over five hours

Test 2d: View the number of items contained within each component order

The order instance and order contents tables split between two tables the tracking of orders placed, and the contents of each order created is likely to be requested commonly, especially for the stores team. However, the query created could be useful for all staff owing to the straightforward fashion that it shows order placed and their size.

The SQL query depicted below selects the 'Order_Component' table and captures the 'Order_ID' and 'Component_ID' fields, applying the COUNT(field_name) function to the latter in order to sum the number of components that are placed against each Order ID instance.

```
--no of items contained in each order
SELECT Order_ID AS [Order_Number], COUNT(Component_ID) AS No_Of_Items_In_Order
FROM Order_Component GROUP BY Order_ID;
```

RESULTS		
	Order_Number	No_Of_Items_In_Order
1	1	3
2	2	1
3	3	2
4	4	2
5	5	1
6	6	1
MESSAGES		
12:19:23	Started executing query at Line 48 (34 rows affected)	
	Total execution time: 00:00:00.035	

Figure 36: Each Order ID and the COUNT() of its component contents

To further extend this query and use a table JOIN clause the field 'Order_Number' could itself be placed inside a COUNT() clause to count the number of orders placed for each Maintenance Job instance. Furthermore, the calculated field 'No_Of_Items_In_Order' could be configured to total the number of items enclosed in all of these orders. This table would then show the number of orders and the number of components for each order.

ii. Timebox 2: Iteration 2

Start Date: 04 April 2019

Anticipated End Date: 08 April 2019

Actual End Date: 10 April 2019

a) Contents

- View all power car jobs in the maintenance schedule
- Add a new power car job into the existing maintenance schedule
- Update the details of an existing power car job in the schedule

b) Feature Design and Implementation

On the screen page a new 'Form' type screen was created, being linked to the database table 'Maintenance_Job' to allow users to view existing table entries, alter existing jobs and add new jobs to the list.

View all power car jobs in the maintenance schedule

The form allows the user to bring up the details of existing records by searching for and selecting the desired Order ID from the "Select a Maintenance Job" search box. Located right at the top of the page, this control provides the user with the spinner with which to select a job from the current work list. By clicking on the spinner each Job ID appears in the list, enabling the user to scroll view each job. Whilst performing this action the other fields present on the form populate with the corresponding job details, as can be seen on a smartphone (iPhone SE) in Figure 37. Albeit different to the view functionality in the wireframe designs, this allows a viewer to easily view the details of any job at a glance, the precise suitability of which will be evaluated further in the prototype heuristic evaluation found in **section 5.2**.

The image displays two mobile application screens side-by-side, representing different record views. Both screens have a light gray header area with a dark green border. The left screen shows a record with Job Vehicle TOPS ID 43140, Vice Vehicle 43087, and Depot LA. The right screen shows a record with Job Vehicle TOPS ID 43012, Vice Vehicle 43027, and Depot LA. Below the header area, there is a 'Select a Job ID' spinner and a 'Done' button. The spinner has a list of numbers: 1, 2, 3, 4, 10, 11, 12, 13, 14. The 'Done' button is located at the bottom right of the screen.

Figure 37: Different Record Views

The fields of type 'DateTime' have been altered to display the time horizontally using the preferred method of a spinner control. This brings up a separate view for date and then time having clicked "Done". **Figure's** Figure 38 and Figure 39 show this control on the iPhone SE and a tablet device (the iPad Mini), the latter retaining the same control size but placing it adjacent to the destination field.

42067

Depot
Location of maintenance job

LA

Due Stop Date/
Time of Vehicle
The time the vehicle
was due to arrive on site
25-03-2018 08:00

Choose a Date
Done

22

December2015

23

January2016

24

February2017

25

March2018

26

April2019

27

May2020

28

June2021

42067

Depot
Location of maintenance job

LA

Due Stop Date/
Time of Vehicle
The time the vehicle
was due to arrive on site
25-03-2018 08:00

Choose a Time
Done

05

57

06

58

07

59

08

00

09

01

10

02

11

03

Figure 38: The Date/Time picker spinner

Job Vehicle TOPS ID
The vehicle receiving maintenance

42092

Vice Vehicle
Add in the TOPS ID of outgoing vehicle if applicable

42067

Depot
Location of maintenance work

LA

Due Stop Date/Time of Vehicle
The time the vehicle was due to arrive on site
25-03-2018 08:00

Actual Stop Date/Time of Vehicle
The time the vehicle actually arrived on site

Due Start Date/Time of Job
The time the exam and/or maintenance was due to start

Actual Start Date/Time of Job
The time the exam and/or maintenance actually commenced

Due End Date/Time of Job
The time the exam and/or maintenance was due to completion

Actual End Date/Time of Job
The time the exam and/or maintenance was actually completed
29-03-2018 08:30

Due Return Date/Time of Vehicle
The time that the vehicle was due to return to service
29-03-2018 11:30

Actual Return Date/Time of Vehicle
The time the vehicle actually returned to service
30-03-2018 06:15

Total Job Length
Total hours taken on the exam and/or maintenance
190.0

Choose a Date
Done

22

December2015

23

January2016

24

February2017

25

March2018

26

April2019

27

May2020

28

June2021

Figure 39: The iPad screen showing the date pickers

Add a new power car job into the existing maintenance schedule

The task of adding in a new record to this table works by leaving the Job ID field blank. This empties all the form fields, allowing the user to add in new details and save the record as a new work piece; a view of which can be seen in *Figure 40*. Data entry is by means of the date and time spinners, option pickers and text box fields.

Select a Maintenance Job *
Search for a record to edit or leave this field blank to create a new record

▼

Job Vehicle TOPS ID
The vehicle receiving maintenance

Enter text...

Vice Vehicle
Add in the TOPS ID of outgoing vehicle if applicable

Select a Job ID Done

1
2

Figure 40: Blank ID to create a new record

One particular challenge that the developer had to overcome was the precise lookup of foreign key fields. When creating or editing records, several fields in the form reference a separate table to provide the user with a definition table lookup, rather than the order of the given field in the linked table. This can be seen in *Figure 41* and *Figure 42* where the control references the 'Staff ID' field in the 'Staff' table and the 'Maintenance_Job' table respectively, the latter omitting the '9' because it already appears in the lookup list, hence referencing the 'Staff' definition table provides a more structured solution.

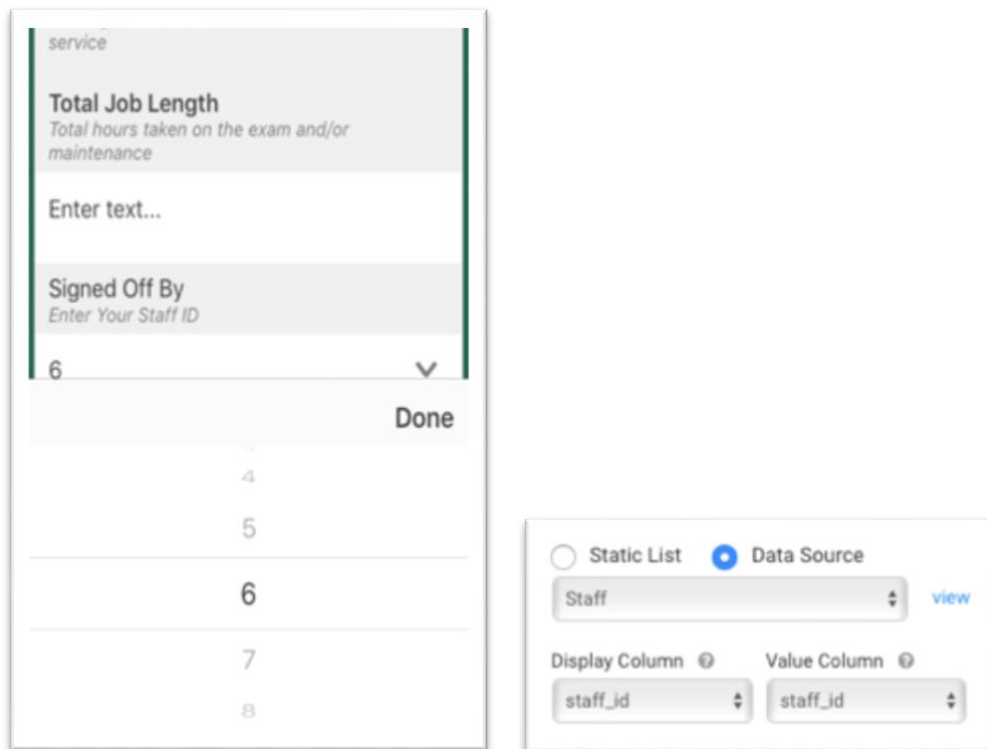


Figure 41: Pulling from the Staff table and the Primary Key field

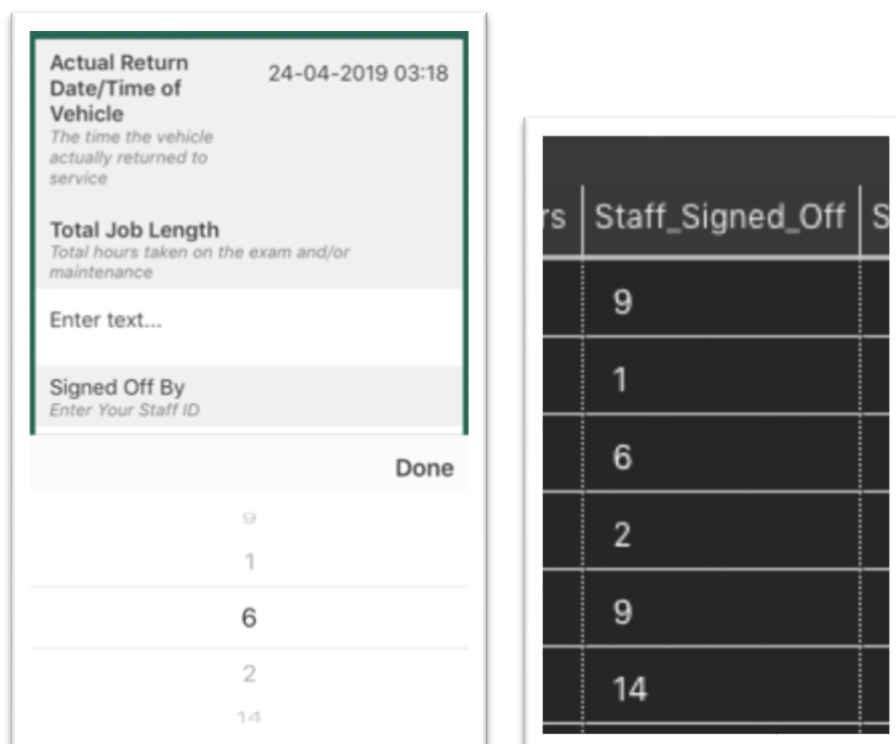
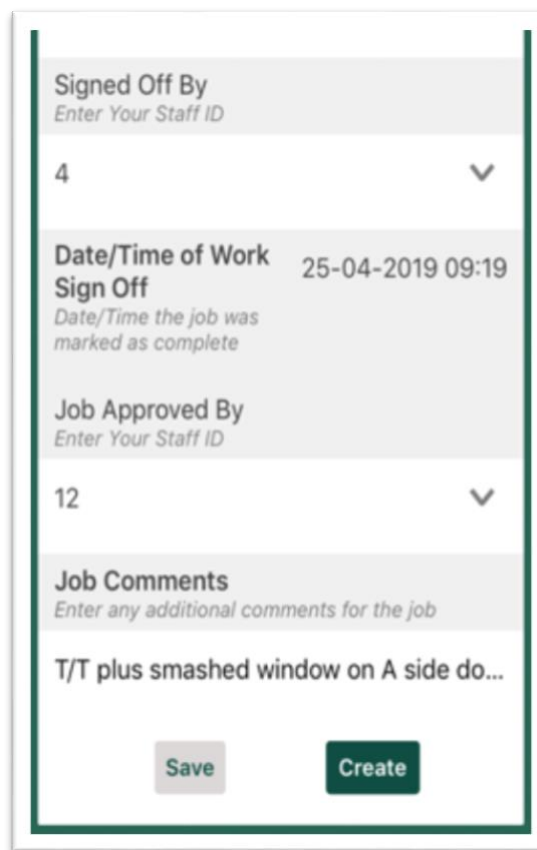


Figure 42: The Staff ID field showing IDs in each of the rows of the Maintenance Job table in the app and table views

Having added the details of a new job to the maintenance schedule, the user can now submit the form to formally create the work item. The form provides the user with the option to “Save” the form whenever they have completed a new record (a new record which is seen in part in Figure 43), finished making adjustments to an existing record, or whenever they are part-way through making changes. Clicking “Save” returns to the “Home” screen and provides confirmation of the save (as shown in Figure 44). A drawback of the tool here is the apparent inability to save the changes but remain on the page. Then, when clicking on the screen icon users are presented with the choice of viewing saved records or creating another new record by viewing the jobs page (as seen in **Figure’s** Figure 45 and Figure 46). Opening up a saved record presents the user with an “Create” option, allowing them to sync changes with the Appenate server and dashboard.



Signed Off By
Enter Your Staff ID

4

Date/Time of Work Sign Off 25-04-2019 09:19
Date/Time the job was marked as complete

Job Approved By
Enter Your Staff ID

12

Job Comments
Enter any additional comments for the job

T/T plus smashed window on A side do...

Save Create

Figure 43: Some fields of a new record before clicking "Save"

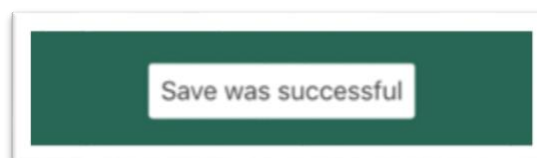


Figure 44: Changes to the record, or additions to a new record are returned as successful

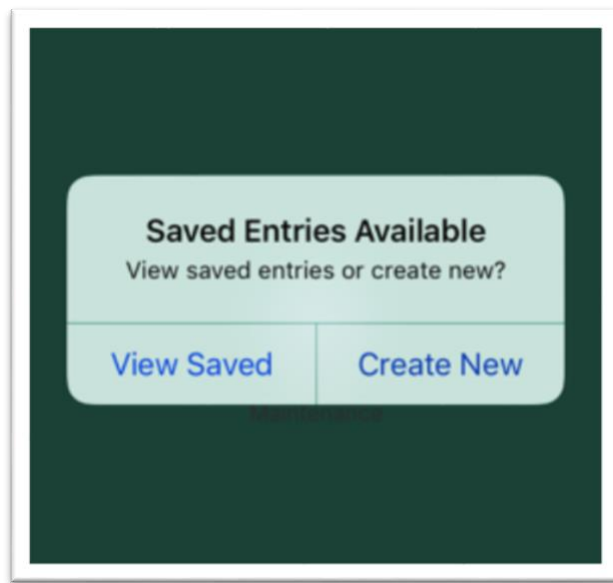


Figure 45: Viewing saved changes to, or new records

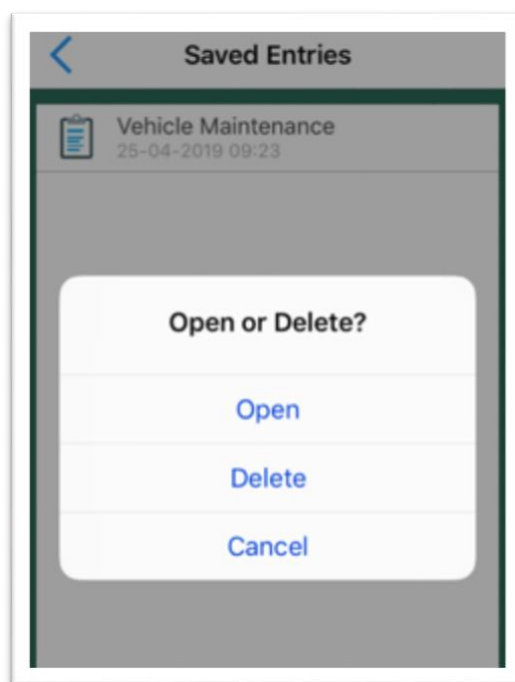


Figure 46: Clicking on a changed or new record to open or delete the changed record

The form creation works with users able to create new records, save them and then reopen them. However, uploading the form to the server continually returned an updated record view containing none of the updated fields, only 'DateTime' fields showing the date and time the user entered into the first date time field on the form, as shown below in the screen captures. The same

erroneous upload also occurred when a user attempts to upload a form without having saved it and reopened it beforehand.

GWR

Vehicle Maintenance

Completed By

Cobbold, Henry

Completed On

25-Apr-2019 11:01

Due Stop Date/Time of Vehicle

25-Apr-2019 11:01

vehicle_due_stop_datetime [Edit](#)

Actual Stop Date/Time of Vehicle

25-Apr-2019 11:01

vehicle_act_stop_datetime [Edit](#)

Due Start Date/Time of Job

25-Apr-2019 11:01

exam_due_start_datetime [Edit](#)

Actual Start Date/Time of Job

25-Apr-2019 11:01

exam_act_start_datetime [Edit](#)

Figure 47: An example of one erroneous record capture

However, this feature is clearly inconsistent as the syncing of the new record was periodically successful, as seen here where the vehicle ID added to a new job has been captured correctly; although the date fields remain erroneous.

Completed By	Cobbold, Henry
Completed On	25-Apr-2019 11:38
Job Vehicle TOPS ID	
	43140
Due Stop Date/Time of Vehicle	
	25-Apr-2019 11:37
Actual Stop Date/Time of Vehicle	
	25-Apr-2019 11:37

Figure 48: This new record sync was partially successful

Update the details of an existing power car job in the schedule

To update the existing records the user must select the desired order ID from the spinner. Now, a user can select any field and change the information contained within. The user must save the record again to keep the changes before a record can be uploaded to the server. Having made, saved and uploaded the changes to a record (in this instance the addition of some work comments seen in Figure 49), the change is returned to the “Data” page of the Appenate dashboard, as captured in Figure 50. “View Entry” shows all the fields of the edited record, as shown in Figure 51 and Figure 52.

5.5

Signed Off By

Enter Your Staff ID

1

▼

Date/Time of Work Sign Off

27-03-2018 08:25

Date/Time the job was marked as complete

Job Approved By

Enter Your Staff ID

12

▼

Job Comments

Enter any additional comments for the job

Replaced windscreen on side A door

Save

Create

Figure 49: Comments appended to Job ID 2

☐	Completed	↓	Completed By		Location		job_id
☐	25-Apr-2019 12:24 Uploaded: 25-Apr-2019 12:24 view entry • export • logs		Cobbold, Henry		51°37'1"N 3°55'34"W		2

Figure 50: The change has synced back to the Appenate server

GWR
Vehicle Maintenance

Completed By	Cobbold, Henry
Completed On	25-Apr-2019 12:24

Select a Maintenance Job	
2	job_id Edit
Job Vehicle TOPS ID	
43140	job_vehicle_no Edit
Vice Vehicle	
43087	vice_vehicle_no Edit

Figure 51: The updated form view

The field view above shows the login user who submitted the changes and at what time. *Figure 52*, below, shows the updated comments field in the job column. This action would allow any signed-in staff member or member of the administrative team to view the changes staff are making to records in 'real time'.

Date/Time of Work Sign Off	
27-Mar-2018 08:25	signed_off_datetime Edit
Job Approved By	
12	staff_approved_job Edit
Job Comments	
Replaced windscreen on side A door	job_comments Edit

Figure 52: The changes made to the comments field

iii. Timebox 3: Iteration 3

Start Date: 10 April 2019

Anticipated End Date: 16 April 2019

End Date: 17 April 2019

a) Contents

- View all orders for components placed
- Add a new order to the list of orders placed
- Update the details of an existing order in the order list

b) Feature Design and Implementation

View all orders for components placed

This screen works in the same way as that discussed throughout timebox two as it allows a user to view the existing records by cycling through the record identified spinner, leaving it blank to allow the creation a new order instance. Updating an existing order to alter dates, comments and status also involves the same processes that have been previously discussed in the preceding development time frame. The “Component Orders” form shows the list of components contained within each order instance.

However, given the foreign key composition of this table the order number appears multiple times in instances where multiple components have been added to an order. In some cases, as seen below in *Figure 54*, numerous components have been added to order numbers 8, 20 and 28. Owing to the need to allocate multiple items to a single order, the order number (against which items can be placed), must be replicated in the database table. This method however, whilst workable on the smartphone, does not translate well onto the tablet where the results returned can occupy the entire screen; as illustrated in *Figure 55*.

To accommodate this in the application a lookup box allows the user to check each record where the order ID is that as specified by the user. The “design view” of the form is shown in *Figure 53* whereby the text box must contain each component corresponding to all the selected records that were checked in the searchable lookup field.

Order Contents

page1

page1

Create or Edit Order Contents

Search for a record to edit or leave this field blank to create a new record

VVV

WWW

XXX

Component ID

Component(s) contained in this order

XXX X XXXX XXXXXXXXX

Comments on Order Contents

Additional notes on order item(s)

XXX X XXXX XXXXXXXXX

Figure 53: The form builder view for Order Contents Screen

8

8

8

18

28

28

28

Component ID

Component(s) contained in this order

114-2F|1602/854200

Comments on Order Contents

Additional notes on order item(s)

I

Create or Edit Order Contents

Search for a record to edit or leave this field blank to create a new record

20

20

20

20

Component ID

Component(s) contained in this order

0015/003810|00341-30|0017/070030

Comments on Order Contents

Additional notes on order item(s)

II

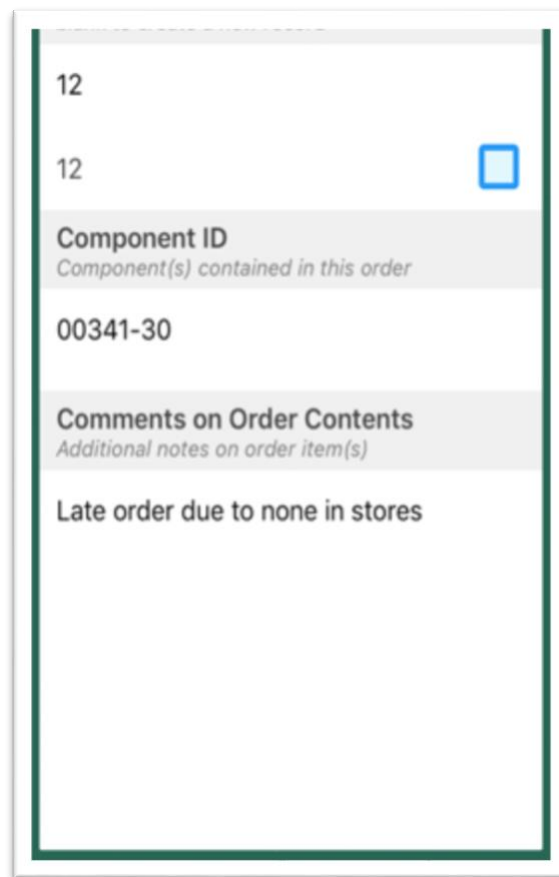
Figure 54: Foreign Key-based table lookup

2	☒
12	☐
20	☐
20	☐
20	☐
21	☐
21	☐
21	☐
21	☐
22	☐
22	☐
22	☐
22	☐
23	☐
24	☐
24	☐
24	☐
24	☐
25	☐

Figure 55: The large results list that can be displayed on an iPad

Add a new order to the list of orders placed

Creating a new order is performed on the same Order Contents page with the user needing to type into the Order ID search box the desired Order ID, at which point, as previously mentioned, the form will display the records from this table matching the Order ID specified. In order to add a new component to Order 12, as illustrated below in *Figure 56*, no order was checked so as to enable a user to add in a new item, in this case component “00341-30” because, as can be seen in *Figure 59*, checking the single component contained in this order displays the ID in the following item text box.



12

12 ☐

Component ID
Component(s) contained in this order

00341-30

Comments on Order Contents
Additional notes on order item(s)

Late order due to none in stores

Figure 56: Creating a new item in Order ID 12

However, just entering the desired Order ID and component as depicted does not fully push the record back to the server, as can be seen in *Figure 57*. The consequence of this inaccurate functionality is that an Order ID search result must be selected in order to push the record back to Appenate. This means that a new component item cannot be placed against an existing order instance defined in the “Component Order” screen from the mobile application.

The screenshot displays the 'GWR' logo at the top, followed by the title 'Order Contents'. Below this, there are two rows of information: 'Completed By' with the value 'Cobbold, Henry' and 'Completed On' with the value '26-Apr-2019 09:47'. A section titled 'Component ID' contains a text input field with the value '00341-30'. To the right of this field is a link labeled 'component_id Edit'.

Completed By	Cobbold, Henry
Completed On	26-Apr-2019 09:47
Component ID	00341-30 component_id Edit

Figure 57: An incomplete upload in the Foreign Key field

Update the details of an existing order in the order list

As alluded to above, a user must select one record in the Component Order to edit the corresponding component item in the order. This can be seen in *Figure 58* where Order 13 has two requested components, forcing the user to select a single record to adjust. This issue can be seen in the database structure where, due to the foreign key composition, only a single component change can be made at a time, as seen in *Figure 58*. Furthermore, to change a particular item the user must check all boxes of that Order ID in order to identify the part to change. When selecting multiple Order ID records, the 'pipe' symbol ("|") is used as the separator, as illustrated in *Figure 54 (right)* where three components comprise Order 20.

Create or Edit Order Contents

Search for a record to edit or leave this field blank to create a new record

13

13

13

☒

☐

Component ID

Component(s) contained in this order

114-2F

Comments on Order Contents

Additional notes on order item(s)

Enter text...

Figure 58: Editing a component part of Order ID 12

Having uploaded the altered record to the Appenate server the following record entry is displayed with the Order ID and ID of the new order item. This view can be seen in *Figure 59*.



Order Contents

Completed By

Cobbold, Henry

Completed On

26-Apr-2019 09:54

Create or Edit Order Contents

13

order_id [Edit](#)

Component ID

114-2F

component_id [Edit](#)

Figure 59: Editing an existing order to alter its Component ID

5. SYSTEM TESTING AND EVALUATION

The FDD approach states that testing should take place at the end of each timebox so that the developer can see if the feature is working as anticipated. Full usability-based system testing of the system will now take place to understand the relatively usability of the initial interface designs and the first prototype application.

5.1 Initial Wireframe Design Testing

To comprehensively test the initial system interface prototypes a combination of a heuristic evaluation and usability test process will be carried out [88]. Due to the busy nature of both the developer and the client, the usability testing process cannot be considered totally representative, as inferred in places throughout this section.

5.1.1 Heuristic Evaluation: Cognitive Walkthrough

For each Personas associated User Story a walkthrough of the steps required to fulfil the user's goal can now be performed. This is often performed before user testing so as to *"clean up"* the interface of fundamental design errors, such as missing control buttons or navigational inconsistencies. This process will help to make subsequent user testing more successful as it will help to allow *"users [to] focus on the experience of pursuing their goals"* [181] during task-based user tests. To perform a Cognitive Walkthrough an initial list of tasks that various users are likely to want to perform will be required. These have been developed from the user stories (or scenarios) that document how these tasks are currently performed – as discussed in **section 4.3.2**. The system evaluators will carry out the same tasks that are likely to be performed by the system users so as to identify inconsistencies from the developer's perspective [182]; an understanding of which can be gained by asking four questions of a user at any given moment:

Q₁: Does the user know what needs to be done? Is the user aware of what they want or need to do? [183],[184]

Q₂: Can the user see a control to do that? Is there anything visible that the user feels can help them perform this action? [183],[184]

Q₃: Is it obvious how the control works? Is it clear how the on-screen visual should be used to achieve successful interaction? [183]

Q4: Does the user understand the feedback they have been given? Can the user make sense of the outcome of the system interaction? [183],[184]

i. Walkthrough of Task A: Marking a Maintenance Job Complete

“Mark has just completed the first task of his shift (changing a power cars traction motor) and must now complete the job’s accompanying paperwork back in the site office before moving onto another piece of work”

In order for the developer to precisely assess a user’s progress towards the particular goal, the individual steps and paths of interaction must be individually evaluated. In order for a user to fulfil the goal outlined in task A the following steps must be taken by the user in the application.

- 1. Sign in with “Staff ID” and “Password”*
- 2. Select the Power car vehicle pane*
- 3. Select (if suitable) the appropriate date filters*
- 4. Select (if suitable) the appropriate exam filters*
- 5. Select (if suitable) the appropriate administrator filters*
- 6. Search for records matching the applied filters*
- 7. Find the correct job from the results list returned*
- 8. Edit the job to complete the empty fields to mark job as complete*
- 9. Save and close the job record*
- 10. Sign out of the application*

Having performed a complete evaluation of each stage, the results of which can be seen throughout **section 10.6**, any tasks that failed to pass the four-aspect test must now be re-developed so as to ensure that subsequent alterations are sufficient so as to pass a second round of cognitive assessment.

Task A Failed Steps:

4. Select appropriate “Exam” filters

Table 9: Task A: Step 4 Initial Test Outcomes

Current Screen View: “Power cars”		
Frame	Heuristic Outcome	Outcome Comments

Does the user know what needs to be done?	Yes	Yes, the user clearly understands if they want to apply any filters or not – all filter controls on this page are optional.
Can the user see a control to do that?	Yes	Yes, the user can see an option to filter results by vehicle and by exam grade in another page “table”.
Is it obvious how the control works?	?	The first field in this design could be misleading as it is shown as a dropdown combo-box. This field is text-based so a searchable dropdown field would improve clarity and suitably show the text-based search. The exam grade field is non-searchable and shows spinner arrows, so is suitable.
Does the user understand the feedback they have been given?	Yes	Yes, the vehicle search control brings up the device keyboard for users to enter the vehicle TOPS ID, typing values brings up search results below the field. Clear and obvious what this action has done. Exam grade field brings up the spinner view at the bottom of the screen so is suitable.

One highlighted solution would be to change the “*Select a Vehicle TOPS ID*” field into a searchable text box. The search control opposed to a spinner allows a quicker entry for a text-based data type. The use of a spinner in the prototype system however will be selected as the developer was ambivalent as to whether a text-based lookup or spinner is clearer for the user. Figure 60 reflects this change in the Balsamiq mock-ups.

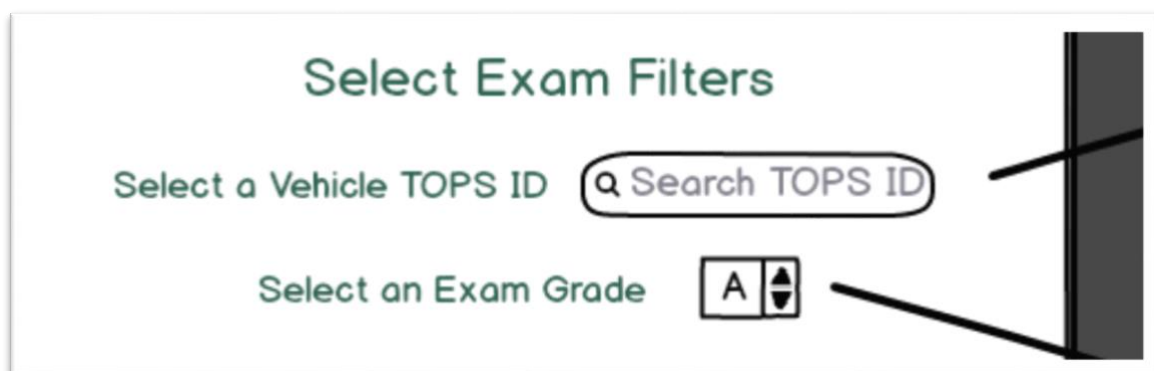


Figure 60: A searchable text box has been added

As can be seen in the corresponding task step shown in *Table 24*, the developer considers the control now more obvious to the user and deems this Walkthrough now to have passed.

5. Select the appropriate “Administration” filters

Table 10: Task A: Step 5 Initial Test Outcomes

<i>Current Screen View: “Power cars”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	They can see that they have the option to select jobs in the schedule by choosing the staff member assigned to them or selecting where the job was carried out.
Can the user see a control to do that?	Yes	Yes, there are clearly several controls for users to set the location and staff using the control label
Is it obvious how the control works?	Yes	Yes, they are both text fields and so are familiar and easily recognisable. The placeholder text also makes this clear.
Does the user understand the feedback they have been given?	No	Initial interface shows a spinner appearing, though this should not appear when using a search bar. Does not show the expected keyboard.

Comparable to that of step 4 of this task, the feedback presented to a user when selecting this control could be misleading. Originally opening the keyboard by means of a spinner wheel icon, the controls now appear as text boxes with the search icon in this section of page. A search box is more commonly associated by users as a control with which to display a keyboard for text input and so is an appropriate solution to this problem. This change can be viewed in *Figure 61*.

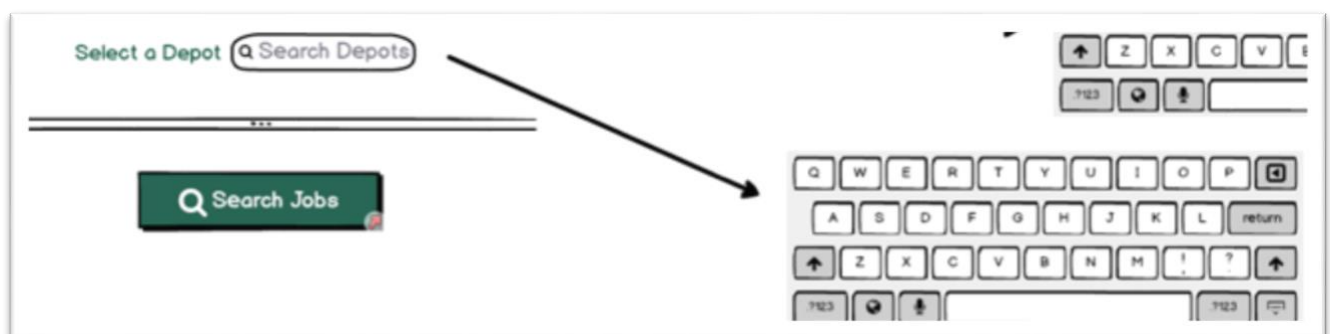


Figure 61: Showing the keyboard

On completion of the discussed changes the step was again placed against the four questions. As shown in *Table 25*, this error is now considered as rectified.

8. Edit the job to complete the empty fields to mark job as complete

Table 11: Task A: Step 8 Initial Test Outcomes

<i>Current Screen View: "Job Details"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, the user understands how to edit the information to update the jobs status.
Can the user see a control to do that?	Yes	Not at the top of the screen but scrolling to the bottom shows the option to <i>"Edit Job"</i> or <i>"Delete Job"</i> . The user wishes to edit so a control is plainly visible here that looks like it will allow this action.
Is it obvious how the control works?	Yes	Yes, it's a button that is widely understood.
Does the user understand the feedback they have been given?	No	Yes and no. This page section shows editable controls including spinners, text fields, searchable boxes and checkboxes (the latter for marking the job complete). It is obvious to the user that each field can now be edited. However, the <i>"Job Orders"</i> segmented control is not suitable because it will not be able to effectively display numerous different orders. Apple suggest that <i>"a segmented control [such as this] should have five or fewer segments"</i> [185].

As noted in *Table 11*, the segmented control currently in this screen section should be replaced. A spinner is a suitable means by which a user can select an item - and importantly provides a suitable means with which to display five or more items with ease and far greater efficiency. *Figure 62*, shows how the item selected in the spinner could be bound to a text field to list the order component IDs.

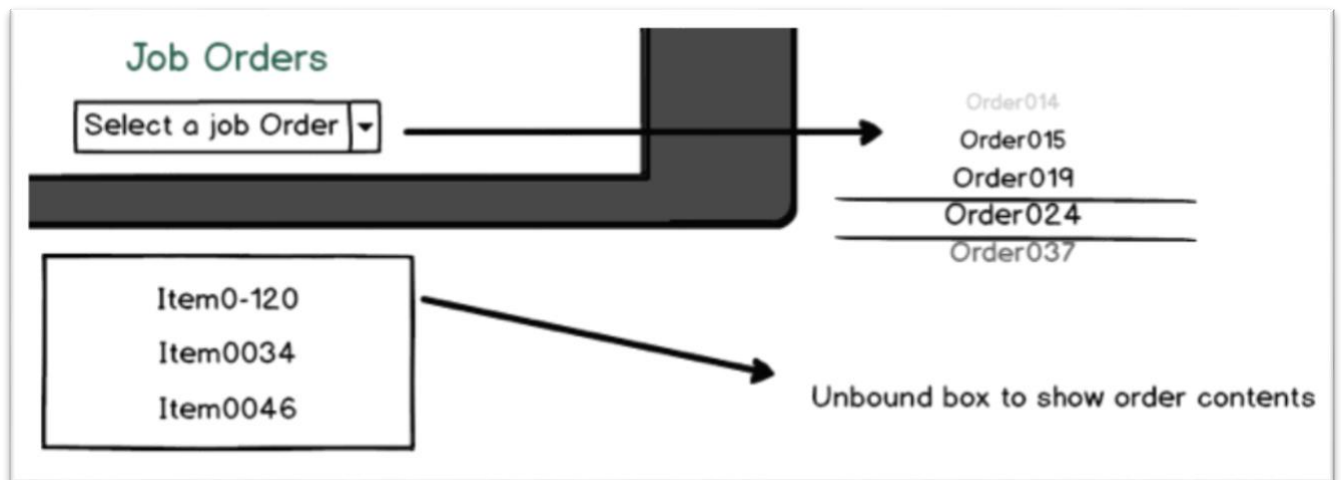


Figure 62: A box now shows contents of spinner order

This change must now be retested to ensure that it passes the developer walkthrough. As can be seen in this step's retest table (*Table 28*) this control accommodates large orders and provides suitable feedback on the contents of an order.

10. Sign out of the program

Table 12: Task A: Step 10 Initial Test Outcomes

Current Screen View: "Job Details"		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	?	This is unclear... Does a user know that they should actively sign out of their account before closing the app? Might they think that simply 'going home' on the device is enough?
Can the user see a control to do that?	Yes	There is a "Sign Out" button visible at the bottom right of the screen. This looks like a way to sign out of an account. The accompanying icon helps make its purpose clear.
Is it obvious how the control works?	Yes	It's a button, so is a widely familiar control.
Does the user understand the feedback they have been given?	Yes	They have returned to the 'Sign In' screen and can see an option to sign into the app. They are aware that this means they have successfully signed out.

The concern that the developer had with this step was that the user might not be aware of an action that they needed to perform. The task of marking a work item as complete should see a

member of staff sign in and then out, on completion, in line with existing IT practice at the site. With *Table 12* noting this as unclear the developer decided that a simple, clear reminder to staff when signing in would be sufficient to remind them to actively sign out, rather than just leave the app and potentially remaining signed in. This change can be seen in *Figure 63* below.

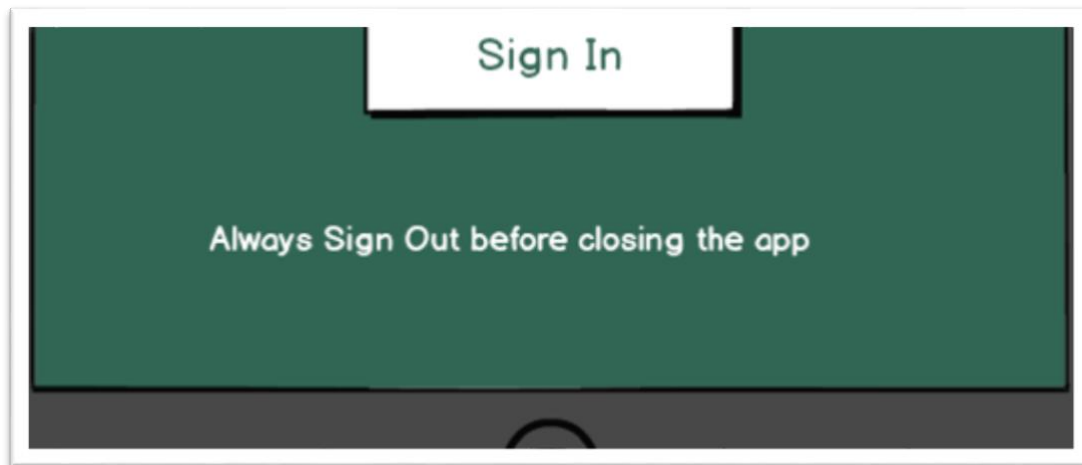


Figure 63: A reminder on the Sign In screen

To demonstrate that this change has been deemed suitable, see *Table 30*.

For the purpose of clarity, for tasks in which this step is present, and also fails, reference is made back to this successful test.

ii. Walkthrough of Task B: Edit an Existing Component Order

“Simon has received an email containing an MRF sheet and must add the necessary components to an existing order because none can be found in the site store. The delivery timeframe for a new order is not viable.”

This task can be broken down into several key steps in the application

1. Sign in with “Staff ID” and “Password”
2. Select the “View Orders” menu pane
3. Select (if suitable) the appropriate “Date” filters to find the given order
4. Select (if suitable) the appropriate “Content” filters to find the given order
5. Select (if suitable) the appropriate “Other” filters to find the given order
6. Search using the selected parameters
7. Find the correct job from the results list returned
8. Amend the appropriate fields to update the orders information
9. Save and close the order record
10. Sign out of the application

Task B Failed Tasks:

3. Select (if suitable) the appropriate “Date” filters to find the given order

Table 13: Task B: Step 3 Initial Test Outcomes

Current Screen View: “Order Schedule”		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	Yes	Yes, there is a sub-heading saying “Select a Date Filter” that instructs the user to enter some date parameters. The user understands that this will help to filter the order results returned.
Can the user see a control to do that?	Yes	Yes, there are two main controls in this “table” (section of page) to define a date range. The calendar symbol helps improve clarity.
Is it obvious how the control works?	?	Unsure. The date pickers in this section of the screen are clear and are widely accepted as date selection tools. However, the checkboxes are not so obvious. Not the suitable control for a tablet due to precision required. A “Yes/No” switch is easier to use and clearer for the user.

Does the user understand the feedback they have been given?	Yes	A date picker spinner appears when clicking on the control. The month, day and year are obviously selectable by scrolling through the wheel. The “Done” button alongside <u>that is part of this control</u> clearly allows the user to exit the spinner having highlighted the desired date. A checkbox (or alternative switch) clearly shows a different state and the result of changing this binary setting.
---	-----	--

As stated, a checkbox is more commonly a web control with which to provide a binary answer from a user. In mobile devices the use of switch-style button controls is far more appropriate and is widely used. Changing checkboxes to switches, in this case the iOS type, is shown below in *Figure 64*.

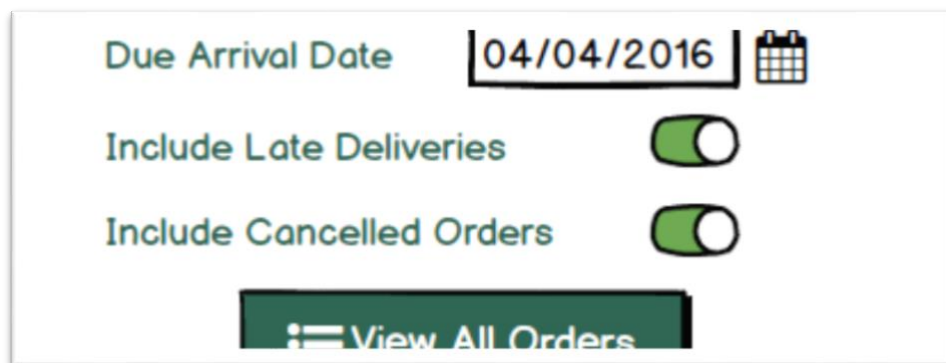


Figure 64: Switches are more suitable

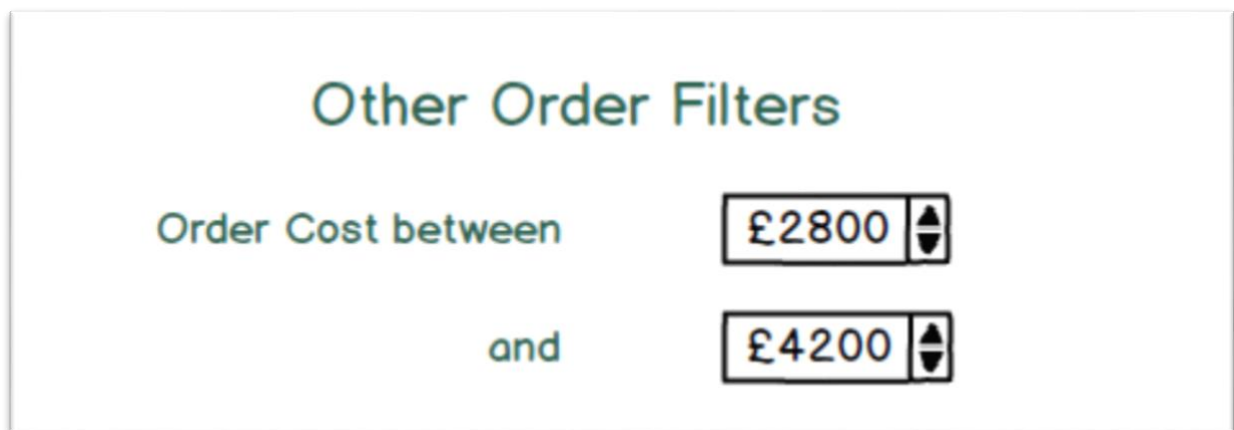
These switch controls have now passed the walkthrough assessment at the second attempt. The retest results are displayed in *Table 33*.

5. Select (if suitable) the appropriate “Other” filters to find the given order

Current Screen View: “Order Schedule”		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	Yes	Again, Yes. The user is aware that they have the option to further refine the order results by looking for precise order cost ranges, orders made by specific staff or orders that have comments appended.
Can the user see a control to do that?	Yes	Yes, several. There are several price spinners and two text field search options with accompanying labels.
	Yes and No	Yes and No. Several fields are suitable, such as both text fields that perform a

Is it obvious how the control works?		search function on orders created by staff or comments added. The field choice for cost is not the most effective. A text field allowing precise number entry, or a stepper, would be more fitting. More accuracy is provided than the series of set increments offered by a spinner.
Does the user understand the feedback they have been given?	Yes	Yes, fields appear with selected contents of the search, or the monetary value from the spinner.

The spinner control present in the designs for number entry is not suitable, and the developer has elected to go with the stepper control. This provides the user with the ability to add a search figure by means of a consistent incremental stepper. This possible change to the initial design is offered in *Figure 65*.



Other Order Filters

Order Cost between £2800

and £4200

Figure 65: Currency steppers rather than spinners

As can be seen in *Table 35*, this change has now passed the Cognitive Walkthrough and can be built into the prototype application interface.

10. Sign out of the application

This issue was highlighted during the Cognitive Walkthrough performed in Task A. Owing to the successful retest of changes made to the interface as part of Task A, this step no longer requires retesting in this Cognitive Walkthrough.

iii. Walkthrough of Task C: Creating a new Maintenance Job

“Ian sees that an attached sheet has been forwarded to him from the site manager and needs to work through the job entries to check for vehicle, capacity or staff issues before adding them to the sites working schedule”.

This task can be broken down into several key steps in the application

1. Sign in with “Staff ID” and “Password”
2. Select the “View Power car Schedule” menu pane
3. View all the power car jobs to view current schedule capacity
4. Go to the “Add New Record” screen in its separate menu
5. Specify the type of record is a maintenance job
6. Create the new maintenance record with desired information
7. Save the newly created job record
8. Sign out of the application

Task C Failed Tasks:

5. Specify the type of record to be added is a maintenance job

Current Screen View: “Add New Record”		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	Yes	They still want to create this new maintenance job in the system.
Can the user see a control to do that?	?	They can see a spinner field underneath the page heading, but other than this it is blank and might be off-putting. This single control seems like something to try to click on.
Is it obvious how the control works?	Yes	It is a combo-box type spinner and is widely used in many mobile interfaces. It should be clear how to use this control.
Does the user understand the feedback they have been given?	Yes	Having selected “Create New Maintenance Job” a range of blank fields and accompanying controls have appeared. Clearly selecting an option from this dropdown has had some effect.

During the walkthrough evaluation for this step the developer felt that users might struggle to understand that they must first select the type of record that they wish to create. A label had been missed from the primary option selector on this page and adding this would provide users with a greater degree of understanding of how to go about creating a new job, order or other table item. The alteration made to add a descriptive label can be seen in *Figure 66* with the subsequent successful retest seen in *Table 45*.



Figure 66: Adding a control caption

8. Sign out of the application

This issue was highlighted during the Cognitive Walkthrough performed in Task A. Owing to the successful retest of changes made to the interface as part of Task A, this step no longer requires retesting in this Cognitive Walkthrough.

5.1.2 Usability Testing

If time is short, such as in this scenario, not all errors identified in the heuristic evaluation or 'expert analysis' necessarily need fixing. One of the benefits of user-based testing is that developers can further their understanding of interface inconsistencies by seeing how users actually respond to them in the system. Beyond fixing the obvious "*low hanging fruit*" [181], some errors identified are supported by user displeasure, a factor that supports their adjustment, yet other errors highlighted might not actually be influencing how the user interacts with a system [181]. However, caution is required as such satisfaction might conceal genuine problems that may require further testing.

This section is split up into three main parts. The first documents the various test scripts used in the user testing process. The second aims to understand the validity of the test participants in the given context whilst the third is the presentation of both quantitative and qualitative results provided by this process.

i. Test Scripts

Owing to the *“demands of evaluating the usability of systems within an industry context”* [186], fast and lightweight methods must be used to maintain timely progress.

Krug’s [187] test script is a tried-and-tested industry standard template offering ample scope for the tester to adjust the script to accommodate different testing scenarios. Fundamentally, the tests included in the process represent the typical tasks performed by accurate system user groups.

Firstly, script starts by gaining an understanding of the user’s site preferences to identify their current software preferences and evaluate their suitability for the usability study. This process is quick and identifies issues in a short period of time so that these can be quickly flagged and processed by *“using [as little as] ... three or four participants”* [188]; ideal given the testing and project timescales [188]. This open nature means that it provides predominantly qualitative results.

To provide more directly quantitative results, the system usability scale (SUS), shown below in *Figure 67*, helps to provide an effective method for *“differentiating between usable and unusable systems”* [189]. Like the Krug [187] framework, this model works well when sample sizes are small. Comparisons can be drawn with the GQM model because these questions also aim to *measure* system usability through which several *goals* can be evaluated. This process occurs by asking users a series of *questions*, as show in **section 2.3.2 ii.**

	Strongly disagree					Strongly agree
1. I think that I would like to use this system frequently	☐	☐	☐	☐	☐	
	1	2	3	4	5	
2. I found the system unnecessarily complex	☐	☐	☐	☐	☐	
	1	2	3	4	5	
3. I thought the system was easy to use	☐	☐	☐	☐	☐	
	1	2	3	4	5	
4. I think that I would need the support of a technical person to be able to use this system	☐	☐	☐	☐	☐	
	1	2	3	4	5	
5. I found the various functions in this system were well integrated	☐	☐	☐	☐	☐	
	1	2	3	4	5	
6. I thought there was too much inconsistency in this system	☐	☐	☐	☐	☐	
	1	2	3	4	5	
7. I would imagine that most people would learn to use this system very quickly	☐	☐	☐	☐	☐	
	1	2	3	4	5	
8. I found the system very cumbersome to use	☐	☐	☐	☐	☐	
	1	2	3	4	5	
9. I felt very confident using the system	☐	☐	☐	☐	☐	
	1	2	3	4	5	
10. I needed to learn a lot of things before I could get going with this system	☐	☐	☐	☐	☐	
	1	2	3	4	5	

Figure 67: Brooke's "Quick and Dirty" System Usability Scale

The benefit of using both methods is that *"testing and inspection find different things"* [181], supporting the strong argument for performing both types of system analysis [181].

Completed usability and SUS scripts can be seen in **sections 10.7** and **10.8** respectively whilst the complete (and unrefined) user testing interviews can be seen in the attached project files.

ii. Participants

Ideally, the participants fit the profiles that have been previously defined in order to test with individuals who most closely fit the systems end-user groups. It is important to suitably select users who have a specific understanding in the target domain, often categorised into 'novice', 'expert' and 'intermediate' users [190].

However, given the minimal client contact this best practice cannot be followed, and it will not be possible to test either the wireframe interface mock-ups or the first prototype product with the end-users of the software. The best alternative is to test the application with naïve users who have no experience with the app, or its intended context - on the premise that they can still find navigational or design inconsistencies [191]. If these users feel that the application navigation methods, form and content structure are of a high quality, then it can be assumed that the real target users will share similar feelings when they use the app in its specific environment - because these factors are broad and can be suitably evaluated by 'everyday' testing groups. The drawback of such conclusions is that the developer will struggle to properly gauge the suitability of the applications context-specific content as any testing process that takes place away from GWR employees cannot answer this question.

iii. Key Findings

a) Usability Testing Script

This section evaluates the key findings of the usability testing process undertaken by three participants. By processing the recorded testing sessions, key issues raised by each participant can be discussed and evaluated to identify areas that require further attention or re-designing.

Having completed the user-testing with the initial wireframes, the interviewer should write up the key points and test results into formal documentation. These test reports have been developed using the notes taken by the interviewer during the session and combined with the discussions that took place during each interview, often containing more information on the user experience with the system that the developer did not note at the time. These formal test reports, sent back the development team and used to inform designers on the outcomes of the user testing process, can be seen below in **Appendix 10.9**. They contain all sections of the test; a tour of the site's initial page, the tasks, additional probing and question sheets and the "*wrapping up of the section*" [191].

Following these reports the key points contained within are discussed with reference to the interface that will form the prototype interface and these are then discussed for each user. These can

subsequently be ranked in order of significance to clearly identify the *most* important issues raised and how they can be confronted [191].

Summary of Test Participant A:

1. User A struggled to go “home”; or navigate to a central page of the app to start a new path of interaction.
2. Overall, User A did not at first interact efficiently with the two-pane menu design. They did however clearly become more familiar with the pane and its structure as they progressed through the user tasks section of testing. They used this structure with ease when exploring the application during the ‘probing’ or ‘questions’ stage of the interview.
3. User A didn’t use (possibly because they didn’t see, at least to begin with) the “*Search Jobs*” or “*Search Orders*” button, instead using “*View All Jobs*” and “*View All Orders*” controls. This might be because the mock-ups didn’t allow any editing of the controls to select options - therefore rendering the option to perform a specific search valueless. This could have led to these controls being overlooked.

Summary of Test Participant B:

1. User B stated that they were expecting to see more options on the “*Edit Order Record*” page. They wanted to change a record on the record page without having to click edit, assuming that they could edit any information without having to click a button to do so. As such, the purpose of this edit option and what the user expected, were clearly different.
2. User B enquired about the use of haptics and additional user feedback mechanisms. These were interesting comments and, although identified as future development feature, were made because of the user’s use of devices in their current field of work.

Summary of Test Participant C:

1. User C performed all tasks well but was very clearly unsure when asked to add a new job record. Although aware that the correct method was to use the “*Add New Record*” process, the user struggled to understand the purpose of all the controls on the page - particularly those relating to ‘Order ID’ and the open comments box. They stated that they expected the comments box to contain information regarding the order destination, though this was already an option for the user in the designated page section.

Several of these concerns could have linked to the way that this was presented in the Balsamiq wireframes. The design process and different control states (as suggested by the user) could have influenced their responses to the tasks by increasing their memory load and slowing their comprehension of the system. Due to time constraints the scope to design and test multiple versions for controls, navigation and copywrite has not been possible.

Summary of Main Usability Nuances:

<i>Test User</i>	<i>Issue Description</i>	<i>Importance</i>	<i>Recommended Fix</i>
A	Wanted to navigate “home” when starting a new path of interaction.	Medium	Specific “Home” button throughout or redesign the menu structure.
A	Took a while to learn/understand using the two-pane menu system.	High	Consider other similar approaches when developing the interface for the prototype application.
A	Did not use “Search Jobs” or “Search Orders” button, instead always used “View All” option.	Low	Not of high priority – by scrolling the user can see that they can filter options and then perform a specific search. Other users did not find this an issue.
B	Thought that “Edit Record” would bring up more editable options than the same ones that they were currently seeing. Meant user thought they could edit what they saw without clicking “Edit” first.	Low	This is dependent on the user’s field of expertise and is a quickly learnt behaviour. It is also not advisable to allow each field to be editable without specifying a wish to do so as this could lead to accidental changes.
B	Felt that additional haptics would make the application easier and clearer to use and improve the efficiency of goal fulfilment.	Low	This is an initial system prototype and has already been highlighted as a desirable feature for future development iterations.
C	Having created a new order, the user struggled to understand how the order instance can have its destination added (i.e. destined for a vehicle or for the site stores)	Medium	Developer needs to look at the options for reworking the fields on the orders page. However, the scope for such changes is limited somewhat by the database structure.
C	Having selected the “Create new Record” option, the process of placing an order against a job was unclear and raised questions from the user.	Medium	This issue is partly due to the layout of the “wireframe” designs. Extra annotations and control (and section) labels might be required.

b) System Usability Scale

Having gathered the key points from the user testing process, the feedback gained from the SUS testing sheets need processing. This scoring system provides a directly quantifiable result – the overall degree of system usability. The calculation and discussion in response to each test user now follows.

- Questions 1, 3, 5, 7 and 9 “*score contribution is the scale position minus 1*” [186].
- Questions 2, 4, 6, 8, 10 the score carried forward is 5 minus the score contribution of that user. E.g. 5 minus 4 (“*Agree*” field) scores 1 [186].
- The re-adjusted scores for each question are then added and multiplied (*) by 2.5 to provide a score /100. A system that returns a score of 68 or above is thought to be above average whilst anything under is said to be below average [186]. This however is a broad measure and the results require a deeper and more considered analysis in order to understand the outcomes of the test, given the case study scenario around which this development is based and the significance of the feedback trends.

It is firstly important to understand that “*there will generally be some [statements] where there is a lot of agreement between respondents*”. This is expected because of the nature of the testing and the often-relative proximity of user needs, perspectives and goals of the target users. However there will still be “*statements [that] provide extreme agreement or disagreement*” within the test population [186] that that require extra attention.

Furthermore, question 1, 9 and 10 are difficult to properly understand or evaluate because they are not aimed at these test users. All the interviewer can do is to encourage the user to put themselves in the position of someone who might be using this app for their job. The notation that “*a user with a lot of prior experience will rate an application as more usable*” [192] means that the significance of users responses to these questions will possibly be unrepresentative of the software’s target groups. [186]. This issue is even more visible in this project whereby, thanks to its highly specific nature, there is a large difference between “*the users with the more experience and those with the least*” [192], or, as is the case here, no degree of prior understanding [192].

Despite this clearly problematic trait of the testing process, the SUS results provided are, overall, highly encouraging and largely support the perspective that the interface poses numerous traits of a usable system. A range of key conclusions can be drawn from the testing of this prototype interface mock-up, as concisely summarised below.

- Users all stated that they would feel satisfied if they had to use the application interface frequently, despite, with the exception of the third test participant, very little understanding of the scenario in which it is to be used.
- All users tested did not consider the interface to be unnecessarily complex. This supports the perspective that, although these users do not represent final system users, the wireframes do support goal fulfilment.
- There were mixed opinions on the need for assistance whilst using the interface. Two users stated that they would not need any help to use the interface. Despite mixed feedback, the users completing the SUS form do not represent GWR site staff who will be using such a system frequently - meaning the support available to the target users should be considered far more readily available.
- All users believed the interface did not demonstrate inconsistencies by stating they *“Disagreed”* or *“Strongly Disagreed”* to the statement *“I thought there was too much inconsistency”*. This metric is of value as the attributes of inconsistency and poor site continuity are often exhibited in sites that offer a poor degree of user satisfaction and goal fulfilment.

5.1.3 Summary of Wireframe Testing

This phase of user testing has clearly been a valuable process, but on reflection it has been identified that the script was not as useful as it could have been. The developer should have adjusted the first set of questions, those originally proposed by Krug [187], to more accurately understand the suitability of the user for the test process. By asking the user about their usage and preferences of ordering-based mobile apps, comparable to those such as online shopping and their use of scheduling based mobile applications (such as calendars) the suitability of the tester for the given scenario can be more accurately gauged. Being able to perform user testing on a larger scale would also have given the developer the option to ‘weed out’ users who have less experience with these apps, increasing the usefulness of opinions held by users that regularly shop online or use scheduling apps on a smartphone.

Based on these views the developer can now change several questions asked during the second round of testing, as well as making changes to the interface based on the feedback discussed throughout **section 5.1.2. iii.**

5.2 Prototype Mobile Application Testing

5.2.1 Heuristic Evaluation: Cognitive Walkthrough

A second Cognitive Walkthrough evaluation will now be performed on the first prototype system. Undertaking a second evaluation is of considerable significance in this project owing to the considerable differences between the interface design of the Balsamiq mock-ups and the front-end options offered by Appenate.

i. Walkthrough of Task A: Marking a Maintenance Job Complete

“Mark has just completed the first task of his shift (changing a power cars traction motor) and must now complete the job’s accompanying paperwork back in the site office before moving onto another piece of work”

Tasks the user must take to achieve this goal using this application.

1. Open the Application
2. Sign In using Appenate credentials
3. Select the “Vehicle Maintenance” menu item
4. Select the correct Maintenance Job ID
5. Update the various screen fields as desired involving scroll
6. Sign off job in correct field
7. Click “Save” to save updated record
8. Select the “Vehicle Maintenance” menu item
9. Select the “View Saved” option
10. Select the correct edit instance from the returned list of edit instances
11. Click “Open” to view it
12. Click “Create” to upload the record
13. Go “Home” to sign out of the application

Task A Failed Steps:

5. Update the various screen fields as desired involving scroll

Table 14: Task A: Step 5 Test Outcomes

Current Screen View: “ Vehicle Maintenance ”		
Frame	Heuristic Outcome	Outcome Comments
	Yes	Yes, they still wish to make changes to the record that they have now selected.

Does the user know what needs to be done?		
Can the user see a control to do that?	Yes, many.	Yes, they can see dropdown icons, date fields and text boxes which provide them with controls to edit the data contained in the record.
Is it obvious how the control works?	Yes and No	The dropdown boxes are clear and work in the same way as the previously used Job ID field, so it is expected there will be no issues in this regard. The DateTime fields do not have an obvious control – i.e. there is nowhere apparent to click. However, just tapping the field clearly displays the date picker spinner so it is anticipated that, despite no icon, a user will have no issue interacting with this control. Text boxes show the filler “Enter Text” so it is clear that these require text entry from a keyboard that will appear.
Does the user understand the feedback they have been given?	Yes	The action of changing any of these fields is clear and obvious to a user.

The nature of the work practised by the vehicle maintenance team on-site means that they work on a 24-hour schedule. This error, identified in *step 5.*, does not make it easy to distinguish between AM or PM. The user has to click on the spinner to see the precise time. This fix however could not be made owing to the limited options provided by the front-end builder in Appenate and no option available to change the display format of this field.

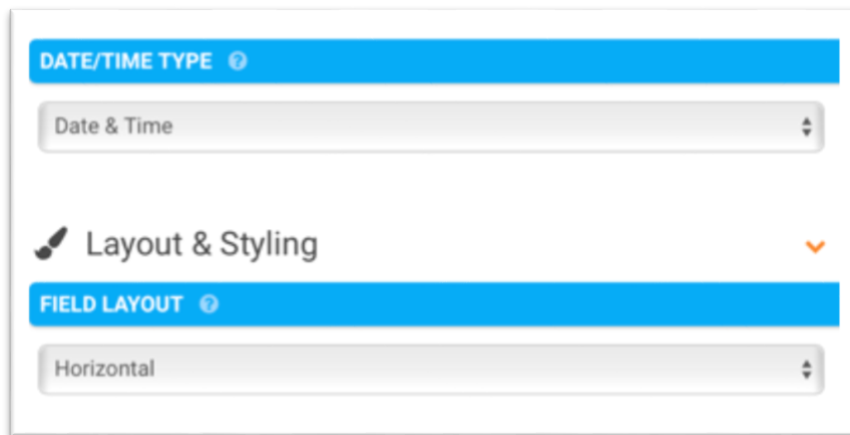


Figure 68: The 'DateTime' fields display options provided are very limited

For more information on how this control discrepancy cannot be corrected please see Table 14.

8. Select the “Vehicle Maintenance” menu item

Table 15: Task A: Step 8 Initial Test Outcomes

<i>Current Screen View: “Home”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	?	The user might not be sure. The record has been successfully saved, but will they understand that this is different to uploading the changes back to the Appenate server? Possibly not. If a user is unaware of this distinct difference the developer believes it is unlikely that they will perform the subsequent steps of this task.
Can the user see a control to do that?	Yes	Yes, a button reading “Vehicle Maintenance” is clearly evident.
Is it obvious how the control works?	Yes	Yes, its unmistakably a button.
Does the user understand the feedback they have been given?	Yes	Yes, they are taken to a page that is showing a completely different view.

This test was highlighted by the Cognitive Walkthrough because the developer is unsure of the two-step “save” and “upload” process. It took the developer a short period of time themselves to

become familiar and understand how changes made to an existing record can be uploaded to the Appenate platform dashboard. As stated in the Cognitive Walkthrough clicking “Upload” returned highlight inconsistent results, the reliability of which seemed to improve when an altered record is saved before it is pushed to Appenate.

This precise functionality however could cause problems as a user is unlikely to be aware, at least during first use, that a record should at first be saved before it can be uploaded; a process that involves closing and re-opening altered records. This clumsy functionality was explored at length by the developer across the Appenate platform but continually provided incorrect data capture of new records. This issue is of considerable importance and would be contained timeboxes that aim to fix application bugs or, such as in this case, errors with Appenate’s record capture. The corresponding retest table can be seen in *Table 75*.

ii. Walkthrough of Task B: Edit an Existing Component Order

“Simon has received an email containing an MRF sheet and must add the necessary components to an existing order because none can be found in the site store. The delivery timeframe for a new order is not viable.”

This task can be broken down into several key steps in the application

1. Open the Application
2. Sign In using Appenate credentials
3. Select the “Component Order” menu item
4. Select the correct Component Order ID
5. Browse the various screen fields as desired involving scroll
6. Exit, not saving any changes made (none have been made)
7. Select the “Order Contents” menu item
8. Select the correct order from the Order ID lookup
9. Edit the existing component contained within using the Component ID field
10. Save the changes made to the record and exit
11. Click on “Order Contents” menu item
12. Click “View Saved Entries”
13. Select the correct edit instance from the returned list of edit instances
14. Click “Open” to view it

15. Click “Create” to upload the record
16. Go “Home” to sign out of the application.

Task B Failed Steps:

11. Click on “Order Contents” menu item

Table 16: Task B: Step 11 Initial Test Outcomes

<i>Current Screen View: “Home”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	?	They might not be sure if they have completed the task. Are they aware that they need to upload the changes, or do they think that simply saving them is enough? If a user thinks that saving the change is enough then they might not complete the rest of the task.
Can the user see a control to do that?	Yes	To access this menu again they can clearly see an “Order Contents” button.
Is it obvious how the control works?	Yes	Yes, it’s a button and is clear.
Does the user understand the feedback they have been given?	Yes	The screen view has changed and a new pop-up dialogue box is being shown.

The issue identified by the developer during this task is similar to the discussed in *Table 15*. The structure of Appenat’s form upload system on a form-based screen is rigid and offers little customisability. Whilst it is anticipated the two-step processes with extra saved changes and exit screens will provide an extra layer of user feedback it is difficult to propose how easily users will understand and the new system and how this is reflected in goal fulfilment. This is one such element of the prototype application that desperately needs clarifying in subsequent user-testing activities to see if this is an issue that potential system users also raise. To see the retest table for this failed step reference *Table 88*.

iii. Walkthrough of Task C: Creating a new Maintenance Job

“Ilan sees that an attached sheet has been forwarded to him from the site manager and needs to work through the job entries to check for vehicle, capacity or staff issues before adding them to the sites working schedule”.

This task can be broken down into several key steps in the application

1. Open the Application
2. Sign In using Appenate credentials
3. Select the “Vehicle Maintenance” menu item
4. Select the correct Maintenance Job ID
5. Browse the various screen fields as desired involving scroll
6. Empty the Job ID field to clear all fields
7. Add in the desired fields so as to create the details for a new piece of work
8. Click “Save” to save updated record
9. Click on “Order Contents” menu item
10. Click “View Saved Entries”
11. Select the correct edit instance from the returned list of edit instances
12. Click “Open” to view it
13. Click “Create” to upload the record
14. Go “Home” to sign out of the application.

Task C Failed Steps:

No steps contained solely within step 3 returned “?” or “No” during the Cognitive Walkthrough process. However, issues previously identified by the developer surrounding a user’s goal awareness and progression were identified in *Task 3*. These have also been raised in failed walkthroughs in both Task 1. (**section i.**) and Task 2. (**section ii.**) and discussed.

5.2.2 Summary of Prototype Cognitive Walkthrough

5.3 Conclusion and Evaluation of System Testing

6. CONCLUSIONS AND RECOMMENDATIONS

6.1 Project Summary and Final Conclusions

As highlighted in *section 3.1*, the purpose of undertaking three distinctive research approaches is to allow the triangulation of results; the success, suitability and significance of which can now be discussed alongside the so as to understand the project's overall success.

Section 1 introduced the case study and the context in which the developer aimed to explore the issues of Just-in-Time supply chain management, work scheduling and resource allocation. Understanding the current work pressures on the firm and their use of digital systems to fulfil the highly strategic, inherently fluid maintenance work schedule for their large fleet of railway vehicles was clearly very important. With a good understanding of the case study company, the next aim was to explore the key issues surrounding the Just-in-Time manufacturing and supply chain management methodology. This process first considered the issues surrounding traditional concepts and how the theories and concepts of, above all else, precise material management, formed the foundations of this continual improvement methodology. The tools used to achieve better efficiency have then been discussed, such as that of the Kanban monitoring system and the holistic, omnipresent 'continual improvement' philosophy. This was further discussed in the supply chain management sphere to form a direct link between the original Asian manufacturing model and the needs of contemporary business reliant on a durable and timely component supply process and accompanying tool.

The development and elsewhere has reinforced the design of a tool that could be used to help streamline the use of existing JIT processes in a firm in which the issues of supply chain management, resource allocation and work scheduling are of substantial significance.

Whilst this broad exploration uncovered numerous tools with which to precisely manage material flow, it was the significance of workplace scheduling and resource management that the author believes was of the most value to the project. Widely commended, these themes act as a base on which scheduling tools, such as Kanban, have been developed and applied to industries outside the historical sphere, but which could benefit from greater efficiency in the handling of resources or work scheduling.

By partnering with a large 'real world' company, the developer can accurately apply an understanding of JIT by developing a maintenance and ordering support tool fit for an industry partner.

It has also been necessary, due to the highly specific nature of the case study business examined, to understand how the notions of system usability can also be applied. By undertaking a literature review with a particular focus on usability in a mobile setting (in which the proposed application must sit) a good understanding of usability and how it can be defined was gained. This topic directed the developer toward methods of expert-level usability evaluation, and wider, more rounded usability-centred system testing and evaluation techniques.

Before the construction of a prototype system could begin, the processes of gathering precise requirements and understanding the needs of the user was necessary. This allowed the developer to construct a formal requirement list with which system development could be prioritised and continually re-evaluated to ensure an appropriate timescale.

In terms of usability, the prototype application should be considered a mixed success. The initial wireframes were seen to be reasonably effect when subjected to a developer-level usability evaluation and this process was of value in that it raised concerns and recommended changes to the mock-ups before subsequent user-based testing was undertaken.

The user-testing processes raised some key considerations for the developer regarding design choices made, but the results and findings should be treated with caution. A sample of users who were quite unfamiliar with the issues took part in this process and only one participant had any degree of prior experience in the field of JIT; but still had no understanding of the precise scenario and tasks that they were expected to attempt.

However, despite the highly promising results obtained from this process and the broader perspective this stage of testing offered, the usability of the initial prototype app cannot be considered quite as successful. The limited back-end functionality of Appenate presented challenges that the developer was unable to overcome, and although a promising prototype has been developed, the nature of this platform means that considerable future development, at least with this tool, is restricted.

To concisely define the project outcomes the developer has come up with the following statements, which are comparable to the project's primary objectives.

- A broad understanding of the issues surrounding the development, values and evolution of the JIT paradigm can clearly be seen.

- The issues surrounding system usability and software context, alongside those of the fore-mentioned JIT, have been of significant importance in prototype evaluation.
- The development of a prototype maintenance and order scheduling support tool can, overall, be considered a success, if not totally realised within the confines of this project.

6.2 Recommendations for Further Research

This section looks at documenting areas of the project that the author believes have not been fully addressed within the scope of this piece. It aims to recommend areas of the project that have seen mixed success, perhaps received less consideration, and areas of peripheral importance that the developer believes should be considered in future research and development activities. In this way the project would be more comprehensive, and the software product more valuable.

The topics that the developer did not have time to fully implement, or consider, during the project include the following.

- While the database constructed may be considered a success within the scope of this project, a stronger and more extensive schema would be required if the developer was to use a foundational mobile application environment instead of a front-end builder.
- To aid the improvement of the existing database a deeper understanding of the varied maintenance tasks performed by GWR staff is likely to be necessary, as well as looking more closely at the supply chain issues faced by component contractors, so that a support tool can more efficiently amalgamate and balance the needs of the supplier and the customer. In this respect the author would like to look into integrated SCM systems and how this prototype tool could be extended to exhibit traits of such platforms.
- The decision to build a front-end interface by using a lightweight builder seemed to offer a suitable solution, but it is now accepted that the process of constructing the application from scratch would allow far more customisability and further development options.
- The opportunity to perform extensive user-testing with GWR staff on-site at Laira in order to understand how the app could be improved in the future would be really valuable. This testing activity should take place before any further development iterations are undertaken so that the developer can precisely identify the elements of the application that require improvement before new features are added.

7. REFLECTION

This section allows the developer to step back from the project and discuss, from a personal perspective, how he felt during its progression and how (from his point of view) he reflects on its different sections and overall outcomes.

I felt that the front-end development environment I selected was far from ideal, and this is clearly highlighted in sections 4.4 and 5.2. In numerous places the limited development options restricted the changes available to me when I performed the cognitive walkthrough. For this reason, I look back on the project with some mixed emotions. The form-based structure provided by Appenate was certainly what I was looking to achieve, but elements of its built-in functionality provided challenges which were difficult to overcome.

I also feel that my time management throughout the project – which has taken 8 months from start to finish - was generally good, with the exception of the development methodologies where I spent too long. I overestimated its importance to the project and committed more time to it than was necessary or sensible. Moving it from the body of the project to the appendix reflects a good deal of wasted time and effort.

Looking back at the case study scenario that I selected, I would like to express my gratitude to the GWR team at Laira in Plymouth, but throughout the project I felt that I would have benefitted from more contact with the team “on location” than was possible to achieve. I felt that this rather limited degree of client contact meant that on numerous occasions I was forced into following my instinct of “best alternative option”, instead of being able to make more confident and better-informed decisions. Despite this essential compromise, which impacted a number of sections in the project, the inability to complete all sections in the “best case”, or to the standard for which I had hoped and planned, frustrated me right through the project. This was particularly apparent when I completed initial interface designs because, while I understand that these should have been approved by the software customer, the schedules for me and company contacts meant that this was not a viable option. Given the highly pressurised nature of this company’s business and the well-established practices which provide continuous vehicle operations, there has been little scope up to now to implement such a support tool – so GWR considered the developer’s project an undertaking of considerable potential future value. It would be interesting to be able to consider the benefits to GWR

of support tools such as this with those managers responsible for more strategic planning and development.

I was rather uncomfortable with the testing process too, which felt forced and a little contrived at times because I was pressed into performing usability testing on subjects who were unrepresentative users, and in numbers that were unlikely to provide me with a reasonably comprehensive list of the usability niggles present in the wireframe designs. Notwithstanding the artificial nature of the testing group, I am aware that I need to work on my user testing approaches if I want to broaden my experience and understanding in this area. I know that I need to work on aspects of the user-testing process (such as refraining from questioning a user when they are taking part in test scenarios!). I believe that I can improve my proficiency with practice and further personal reading around this process. I feel that this is attainable because, while this activity felt forced to begin with, I quickly began to relax during the user testing interviews and the final wireframe interview felt far more natural. This too provided, in my opinion, more valuable feedback.

I consider the project overall to be a success. I selected this project because it would test my database development skills and, to challenge me further, allow me to work with areas in the wider range of IT that particularly interest me. This of course meant being prepared to come away from my comfort zone and move into less familiar areas.

To summarise this reflective section, I really enjoyed working on this project and would like to look further into the topic of UX in the future. So, in terms of a first step into a large industrial project in which context and usability are central, I am very happy that I selected this case study. Furthermore, the skills that I've learnt and improved as this long project has progressed are highly transferable. The responsibility for such a large industrial project, being able to perform systematic research and demonstrating, presenting and explaining my thought processes and decision-making will be valuable and important skills and experiences for me for the future.

8. REFERENCES AND CITED WORK(S)

- [1] First Group plc., "First Rail – FirstGroup plc," *A great track record in rail*, 2018. [Online]. Available: <http://www.firstgroupplc.com/about-firstgroup/uk-rail.aspx>. [Accessed: 30-Apr-2018].
- [2] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 6th ed. Boston, MA: Addison-Wesley, 2016.
- [3] UniPart Rail, "Unipart Rail - Supply Chain Solutions," *About Us: Supply Chain Solutions*, 2018. [Online]. Available: <https://www.unipartrail.com/supply-chain-solutions.html>. [Accessed: 12-Feb-2019].
- [4] K. Lyons and B. Farrington, *Purchasing and Supply Chain Management*, 8th ed. Harlow, Essex, United Kingdom: Pearson Education, 2012.
- [5] Y. Sugimori, K. Kusunoki, F. Cho, and S. Uchikawa, "Toyota Production System and Kanban system materialisation of just-in-time and respect-for-human system," *Int. J. Prod. Res.*, vol. 15, no. 6, pp. 553–564, 1977.
- [6] R. J. Vokurka and R. A. Davis, "Just-in-time: The evolution of a philosophy," *Prod. Invent. Manag. J.*, vol. 37, no. 2, pp. 56–58, 1996.
- [7] Supply Chain Visions, "Supply Chain and Logistics Terms and Glossary," *www.scvisions.com*, 2010. [Online]. Available: www.scvisions.com. [Accessed: 12-Nov-2018].
- [8] V. A. Mabert, "The early road to material requirements planning," *J. Oper. Manag.*, vol. 25, no. 2, pp. 346–356, 2007.
- [9] N. R. Sanders, *The Definitive Guide to Manufacturing and Service Operations: Master the Strategies and Tactics for Planning, Organizing, and Managing How Products and Services Are Produced*, 1st ed. Upper Saddle River, New Jersey, NJ: Pearson FT Press, 2013.
- [10] F. Wilson, J. Desmond, and H. Roberts, "Success and Failure of MRP II Implementation," *Br. J. Manag.*, vol. 5, pp. 221–240, 1994.
- [11] W. C. Benton and H. Shin, "Manufacturing Planning and Control: The Evolution of MRP and JIT Integration," *Eur. J. Oper. Res.*, vol. 110, pp. 411–440, 1998.
- [12] A. Jacobi, Michael, "How to unlock the benefits of MRP II and just-in-time," *Hosp. Mater. Manag. Q.*, vol. 15, no. 4, pp. 12–22, 1994.
- [13] J. K. Liker, "The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer," McGraw-Hill, New York, NY, 2004.
- [14] W. J. Hopp and M. L. Spearman, "To Pull or Not to Pull: What Is the Question?," *Manuf. Serv.*

- Oper. Manag.*, vol. 6, no. 2, pp. 133–148, 2004.
- [15] G. Singh and I. S. Ahuja, “Just-in-time manufacturing: literature review and directions,” *Int. J. Bus. Contin. Risk Manag.*, vol. 3, no. 1, p. 57, 2012.
 - [16] N. Radziwill, “The Origins of Just-In-Time,” *Quality and Innovation*, 2010. [Online]. Available: <https://qualityandinnovation.com/2010/10/13/the-origins-of-just-in-time/>. [Accessed: 28-Sep-2018].
 - [17] L. Heiko, “A Simple Framework For Understanding JIT,” *Prod. Invent. Manag. J.*, vol. 30, no. 4, pp. 61–63, 1989.
 - [18] M. Apreutesei, E. Suci, I. R. Arvinte, and D. Munteanu, “Application of Kanban System for Managing Inventory,” *Bull. Transilv. Univ. Brasov*, vol. 3, no. 52, p. 6, 2010.
 - [19] S. J. Pavnaskar, J. K. Gershenson, and A. B. Jambekar, “Classification scheme for lean manufacturing tools,” *Int. J. Prod. Res.*, vol. 43, no. 13, pp. 3075–3090, 2003.
 - [20] V. R. Kannan and K. C. Tan, “Just in time, total quality management, and supply chain management: Understanding their linkages and impact on business performance,” *Omega*, vol. 33, no. 2, pp. 153–162, 2005.
 - [21] H. Cobbold, “Developing an application in the context of the rail Industry based on ‘JIT’ procurement principles: A project proposal,” Swansea, United Kingdom, 2018.
 - [22] Y. Monden, *Toyota Production System: An Integrated Approach to Just-In-Time*, 4th ed. New York, NY: CRC Press, 2011.
 - [23] J. S. Randhawa and I. Singh Ahuja, “5S – a quality improvement tool for sustainable performance: literature review and directions,” *Int. J. Qual. Reliab. Manag.*, vol. 34, no. 3, pp. 334–361, 2017.
 - [24] R. J. Vokurka and R. R. Lummus, “The Role of Just-In-Time in Supply Chain Management,” *Int. J. Logist. Manag.*, vol. 11, no. 1, pp. 89–98, 2000.
 - [25] A. J. Kootanaee, K. N. Babu, and H. F. Talari, “Just-In-Time Manufacturing System: From Introduction to Implement,” *Int. J. Econ. Bus. Financ.*, vol. 1, no. 2, pp. 7–25, 2013.
 - [26] T. Ohno, *Toyota Production System, Beyond Large Scale Production Systems*. Cambridge, MA: Productivity Press, 1988.
 - [27] R. M. Monczka, R. B. Handfield, L. C. Giunipero, J. L. Patterson, and D. Waters, *Purchasing and Supply Chain Management*. Andover: South-Western Cengage Learning, 2010.
 - [28] M. Sahil, “Implications of Just-In-Time System of Toyota: A Case Study,” Kyoto, 2017.
 - [29] A. Natasya, A. Wahab, M. Mukhtar, and R. Sulaiman, “A Conceptual Model of Lean Manufacturing Dimensions,” in *The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013)*, 2013, vol. 11, pp. 1292–1298.

- [30] M. Brown, "Toyota Production System and Supply Chain Management," Zaragoza Logistics Centre, 2005.
- [31] S. Bhasin and P. Burcher, "Lean viewed as a philosophy," *J. Manuf. Technol. Manag.*, vol. 17, no. 1, pp. 56–72, Jan. 2006.
- [32] J. Jayaram, A. Das, and M. Nicolae, "Looking beyond the obvious: Unraveling the Toyota production system," *Int. J. Prod. Econ.*, vol. 128, no. 1, pp. 280–291, 2010.
- [33] M. S. Spencer, D. S. Rogers, and P. J. Daugherty, "JIT Systems and External Logistics Suppliers," *Int. J. Oper. Prod. Manag.*, vol. 14, no. 6, pp. 60–74, 2002.
- [34] N. K. Shimbun, *Poka-Yoke: Improving Product Quality by Preventing Defects*, 1st ed. Cambridge, MA: Productivity Press, 1989.
- [35] J. Pettersen, "Defining lean production: Some conceptual and practical issues," *TQM J.*, vol. 21, no. 2, pp. 127–142, 2009.
- [36] H. Robinson, "Using Poka-Yoke Techniques for Early Defect Detection," 2011.
- [37] S. Shingo, *The Shingo Prize Production Management System: Improving Process Functions*. Cambridge, MA: Productivity Press, 1992.
- [38] R. B. Pojasek, "Poka-Yoke and Zero Waste," *Environ. Qual. Manag.*, vol. 9, no. 2, pp. 91–97, 1999.
- [39] S. Gupta and S. K. Jain, "The 5S and kaizen concept for overall improvement of the organisation: a case study," *Int. J. Lean Enterp. Res.*, vol. 1, no. 1, pp. 22–40, 2014.
- [40] J. Michalska and D. Szewieczek, "The 5S methodology as a tool for improving the organisation," *J. Achiev. Mater. Manuf. Eng.*, vol. 24, no. 2, pp. 211–214, 2007.
- [41] A. Manos, "The Benefits of Kaizen and Kaizen Events," *Qual. Prog.*, vol. 40, no. 2, pp. 47–48, 2007.
- [42] kanbanize.com, "Kanban Explained for Beginners," *Getting Started with Kanban*, 2018. [Online]. Available: <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban/>. [Accessed: 08-Oct-2018].
- [43] D. Y. Golhar and C. L. Stamm, "The just-in-time philosophy: A literature review," *Int. J. Prod. Res.*, vol. 29, no. 4, p. 657, 1991.
- [44] M. S. Akturk and F. Erhun, "An overview of design and operational issues of Kanban systems," *Int. J. Prod. Res.*, vol. 37, no. 17, pp. 3859–3881, 1999.
- [45] D. J. Anderson, *Kanban: Successful Evolutionary Change for your Technology Business*, Blue Hole. Seattle, WA, 2010.
- [46] H. T. Ingason, E. Gestsson, and H. I. Jonasson, "The Project Kanban Wall: Combining Kanban and Scrum for Coordinating Software Projects," *PM World J.*, vol. II, no. VIII, pp. 1–23, 2013.

- [47] C. S. Kumar and R. Panneerselvam, "Literature review of JIT-KANBAN system," *Int. J. Adv. Manuf. Technol.*, vol. 32, no. 3–4, pp. 393–408, 2007.
- [48] F. Giordano and M. M. Schiraldi, "On Just-In-Time Production Leveling," *Oper. Manag.*, pp. 141–162, 2013.
- [49] O. Al-Baik and J. Miller, "The kanban approach, between agility and leanness: a systematic review," *Empir. Softw. Eng.*, vol. 20, no. 6, pp. 1861–1897, 2015.
- [50] D. Mann, "The Missing Link: Lean Leadership," *Front. Health Serv. Manage.*, vol. 26, no. 1, pp. 15–26, 2009.
- [51] J. Bicheno, *The New Lean Toolbox: Towards Fast, Flexible Flow*, 3rd ed. Buckingham: PICSIE Books, 2004.
- [52] A. Grzelczak and K. Werner-Lewandowska, "Eliminating muda (waste) in lean management by working time," *Arab. J. Bus. Manag. Rev.*, vol. 6, no. 3, pp. 1–11, 2016.
- [53] K. Zimmer, "Supply chain coordination with uncertain just-in-time delivery," *Int. J. Prod. Econ.*, vol. 77, no. 1, pp. 1–15, 2002.
- [54] M. Christopher, *Logistics and Supply Chain Management*, 4th ed. Harlow, Essex: Pearson Education, 2011.
- [55] S. Min and J. T. Mentzer, "Developing and measuring supply chain management concepts," *J. Bus. Logist.*, vol. 25, no. 1, pp. 63–100, 2004.
- [56] R. K. Oliver and M. D. Webber, "Supply Chain Management: Logistics catches up with strategy," in *Logistics: The Strategy Issues*, London, UK: Chapman and Hall, 1992, pp. 63–75.
- [57] J. Liu, S. Zhang, and J. Hu, "A case study of an inter-enterprise workflow-supported supply chain management system," *Inf. Manag.*, vol. 42, no. 3, pp. 441–454, 2005.
- [58] C. Stefanou, "Supply Chain Management (SCM) and organizational key factors for successful implementation of Enterprise Resource Planning (ERP) systems," in *Proceedings of the Americas Conference on Information Systems (AMCIS)*, 1999, pp. 800–818.
- [59] J. Oláh, Z. Zéman, I. Balogh, and J. Popp, "Future Challenges and areas of development for supply chain management," *Sci. J. Bus. Logist.*, vol. 14, no. 1, pp. 127–138, 2017.
- [60] M. P. Robillard, "What Makes APIs Hard to Learn? The Answers of Developers," *IEEE Softw.*, vol. 26, no. 6, 2009.
- [61] H. Stadtler, "Supply chain management and advanced planning - Basics, overview and challenges," *Eur. J. Oper. Res.*, vol. 163, no. 3, pp. 575–588, 2005.
- [62] A. Shalloway, "Demystifying Kanban," *Cut. IT J.*, vol. 24, no. 3, pp. 12–17, 2011.
- [63] A. I. Wasserman, "Software engineering issues for mobile application development," in *FoSER (Future of Software Engineering Research)*, 2010, pp. 397–400.

- [64] S. Sarker and J. D. Wells, "Understanding mobile handheld device use and adoption," *Commun. ACM*, vol. 46, no. 12, p. 35, 2003.
- [65] F. Nayebi, J.-M. Desharnais, and A. Abran, "The state of the art mobile application usability evaluation," in *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2012, pp. 1–4.
- [66] T. Liu, X. Wu, P. Sun, and H. Wang, "Frameworks for Exploring the User Experience of Mobile Apps," *2016 Int. Conf. Sustain. Energy, Environ. Inf. Eng. (Seeie 2016)*, no. Seeie, pp. 307–312, 2016.
- [67] K. de Silva, "The 2018 Mobile Marketing Report: Stats You Need To Know," *Mobile Marketing News*, 2018. [Online]. Available: <https://www.biznessapps.com/blog/2018-mobile-marketing-report-stats-need-know/>. [Accessed: 12-Feb-2019].
- [68] S. Burford and S. Park, "The impact of mobile tablet devices on human information behaviour," *J. Doc.*, vol. 70, no. 4, pp. 622–639, 2014.
- [69] C. Banga and J. Weinhold, *Perfecting Interface Design In Mobile Apps: Essential Mobile Interaction Design*. Upper Saddle River, NJ: Addison-Wesley, 2014.
- [70] E. Park and A. P. Del Pobil, "Technology acceptance model for the use of tablet PCs," *Wirel. Pers. Commun.*, vol. 73, no. 4, pp. 1561–1572, 2013.
- [71] D. Zhang and B. Adipat, "Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications," *Int. J. Hum. Comput. Interact.*, vol. 18, no. 3, pp. 293–308, 2009.
- [72] D. Cyr, M. Head, and A. Ivanov, "Design aesthetics leading to m-loyalty in mobile commerce," *Inf. Manag.*, vol. 43, no. 8, pp. 950–963, 2006.
- [73] A. Idri, K. Moumane, and A. Abran, "On the use of software quality standard ISO/IEC9126 in mobile environments," in *2013 20th Asia-Pacific Software Engineering Conference*, 2013, vol. 1, pp. 1–8.
- [74] B. Schneiderman, C. Plaisant, M. Cohen, S. Jacobs, and N. Elmqvist, *Designing the User Interface*, 6th ed. Harlow, Essex: Pearson Education, 2018.
- [75] F. Lachner, P. Naegelien, R. Kowalski, M. Spann, and A. Butz, "Quantified UX: Towards a Common Organisation Understanding of User Experience," *Int. Conf. Des. User Exp. Usability*, no. June, pp. 230–241, 2014.
- [76] N. A. Ismail, F. Ahmad, N. A. Kamaruddin, and R. Ibrahim, "A Review on Usability Issues in Mobile Applications," *IOSR J. Mob. Comput. Appl.*, vol. 3, no. 3, pp. 47–52, 2016.
- [77] International Organisation for Standardisation (ISO), "Ergonomics of human-system interaction - Part 11: Usability: Definitions and Concepts," Geneva, Switzerland, 2018.
- [78] D. Van Greunen, A. Van der Merwe, and P. Kotze, "Factors influencing BPM tools: The

- influence on user experience and user interfaces,” *Int. J. Comput. ICT Res.*, vol. 4, no. 1, pp. 47–57, 2010.
- [79] J. Rubin and D. Chisnell, *Handbook for Usability Testing, Second Edition: How to Plan, Design and Conduct Effective Tests*, 2nd ed. Indianapolis, IN: Wiley Publishing Inc., 2008.
 - [80] N. Bevan, “Classifying and Selecting UX and Usability Measures,” in *Proceedings of the International Workshop on Meaningful Measures: Valid Useful User Experience Measurement (VUUM)*, 2008, pp. 67–71.
 - [81] N. Bevan, J. Carter, and S. Harker, “Iso 9241-11 revised: What have we learnt about usability since 1998?,” in *International Conference on Human-Computer Interaction*, 2015, no. August, pp. 143–151.
 - [82] A. Seffah, M. Donyaee, R. B. Kline, and H. K. Padda, “Usability measurement and metrics: A consolidated model,” *Softw. Qual. Journal; Dordr.*, vol. 14, no. 2, pp. 159–178, 2006.
 - [83] N. Bevan, “Measuring Usability as Quality of Use,” *Softw. Qual. J.*, vol. 4, pp. 1–20, 1995.
 - [84] N. Bevan and M. Macleod, “Usability measurement in context,” *Behav. Inf. Technol.*, vol. 13, no. 1, pp. 132–145, 1994.
 - [85] N. Bevan, “What is the difference between the purpose of usability and user experience methods?,” *Proc. Work. UXEM*, vol. 9, pp. 1–4, 2009.
 - [86] N. Bevan, “Human-computer interaction standards,” *Adv. Hum. Factors/Ergonomics*, vol. 20, no. B, pp. 885–890, 1995.
 - [87] D. Swallow, M. Blythe, and P. Wright, “Grounding Experience: Relating Theory and Method to Evaluate the User Experience of Smartphones,” in *Proceedings of the 2005 annual conference on European association of cognitive ergonomics.*, 2005, pp. 91–98.
 - [88] J. Nielsen, “Usability Inspection Methods,” in *Conference companion on Human factors in computing systems*, 1994, pp. 413–414.
 - [89] P. Wright and J. McCarthy, *Technology as Experience*. Massachusetts, MA: MIT Press, 2004.
 - [90] A. Kankainen, “Thinking model and tools for understanding user experience related to information appliance product concepts,” Helsinki University of Technology, 2002.
 - [91] W. Maehr, Y. Otero, M. Fjeld, and L. E. Bolstad, “Mobile Internet User Experience in the Western World: Cost-Benefit Estimation,” *Mob. HCI 2007 Work.*, pp. 3–6, 2007.
 - [92] J. Nielsen and R. Molich, “Heuristic Evaluation of User Interfaces,” *Proc. SIGSCHI Conf. Hum. factors Comput. Syst.*, vol. March, no. 1, pp. 373–380, 1990.
 - [93] J. Nielsen, “Usability 101: Introduction to Usability,” *Nielsen Normal Group*, 2012. [Online]. Available: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>. [Accessed: 18-Dec-2018].

- [94] A. Abran, A. Khelifi, W. Suryn, and A. Seffah, "Usability meanings and interpretations in ISO standards," *Softw. Qual. J.*, vol. 11, no. 4, pp. 325–338, 2003.
- [95] P. Morville, "User Experience Design," *Semantic Studios*, 2004. [Online]. Available: https://semanticstudios.com/user_experience_design/. [Accessed: 18-Dec-2018].
- [96] O. Boon Jiet and A. Zuhairi Abdul Majid, "Retail Efficiency and Productivity Enhancement Through Ergonomic Design," in *1st International Conference on Creative Media, Design and Technology*, 2014.
- [97] D. Wesolko, "Peter Morville's User Experience Honeycomb," *medium.com*, 2016. [Online]. Available: <https://medium.com/@danewesolko/peter-morvilles-user-experience-honeycomb-904c383b6886>. [Accessed: 14-Jan-2019].
- [98] H. T. Paulsrud, "A discussion of Technological Frames applied at an Usability Test," University of Oslo, 2011.
- [99] A. Hussain and M. Kutar, "Usability metric framework for mobile phone application," *10th Annu. Postgrad. Symp. Converg. Telecommun. Netw. Broadcast.*, no. June 2009, pp. 1–5, 2009.
- [100] M. Satyanarayanan, "Fundamental challenges in mobile computing," *Proc. fifteenth Annu. ACM Symp. Princ. Distrib. Comput. - Pod. '96*, pp. 1–7, 1996.
- [101] J. Gong and P. Tarasewich, "Guidelines for handheld mobile device interface design," *Proc. DSI 2004 Annu. Meet.*, vol. Nov, pp. 3751–3756, 2004.
- [102] C. Ryan and A. Gonsalves, "The Effect of Context and Application Type on Mobile Usability: An Empirical Study," in *28th Australasian Computer Science Conference*, 2005, vol. 38.
- [103] J. Kjeldskov, "'Just-in-Place' Information for Mobile Device Interfaces," *Mob. HCI 2002*, no. 2411, pp. 25–25, 2002.
- [104] W. C. Foundry, W. Altoros, and A. Altoros, "Mobile Application Development Trends and Challenges," *Int. J. Eng. Technol.*, vol. 3, no. 8, pp. 1096–1099, 2016.
- [105] S. L. Jarvenpaa and K. R. Lang, "Managing the paradoxes of mobile technology," *Inf. Syst. Manag.*, vol. 22, no. 4, pp. 7–23, 2005.
- [106] A. Schmidt, "Implicit Human Computer Interaction Through Context," *Pers. Technol.*, vol. 4, no. 2/3, pp. 191–199, 2000.
- [107] S. Nidhra and J. Dondeti, "Black Box and White Box Testing Techniques - A Literature Review," *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, pp. 29–50, 2012.
- [108] D. C. Crowther and P. J. Clarke, "Examining Software Testing Tools," *Dr. Dobbs Journal; San Mateo*, vol. 30, no. 6, pp. 28–33, 2005.
- [109] B. Hailpern and P. Santhanam, "Software debugging, testing and verification," *IBM Syst.*

- Manag.*, vol. 41, no. 1, pp. 4–12, 2002.
- [110] C. Lamsfus, D. Wang, A. Alzua-Sorzabal, and Z. Xiang, “Going Mobile: Defining Context for On-the-Go Travelers,” *J. Travel Res.*, vol. 54, no. 6, pp. 691–701, 2015.
 - [111] T. Rodden, K. Chervest, N. Davies, and A. Dix, “Exploiting Context in HCI Design for Mobile Systems,” *Work. Hum. Comput. Interact. with Mob. devices*, no. May, pp. 21–22, 1998.
 - [112] Y. E. Lee and I. Benbasat, “Interface Design for Mobile Commerce,” *Commun. ACM*, vol. 46, no. 12, pp. 49–52, 2003.
 - [113] A. K. Dey, “Understanding and Using Context,” *Pers. Ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, 2001.
 - [114] U. Alegre, J. C. Augusto, and T. Clark, “Engineering context-aware systems and applications: A survey,” *J. Syst. Softw.*, vol. 117, pp. 55–83, 2016.
 - [115] J. Lowgren, “Interaction Design,” in *Encyclopedia of Human Computer Interaction*, 2nd ed., The Interaction Design Foundation, 2007, pp. 7–18.
 - [116] M. Poli and A. J. Shenhar, “Project strategy: the key to project success,” no. August 2003, pp. 231–235, 2004.
 - [117] P. Verschuren and H. Doorewaard, *Designing a Research Project*, 2nd ed. The Hague, The Netherlands: Eleven International Publishing, 2010.
 - [118] N. Whitehead, “Research Methods II.” UWTSD, Swansea, United Kingdom, 2012.
 - [119] D. E. Gray, *Doing Research in the Real World*, 3rd ed. London, UK: Sage Publications, 2014.
 - [120] J. Dudovskiy, “Deductive Approach (Deductive Reasoning),” *Research Methodology*, 2019. [Online]. Available: <https://research-methodology.net/research-methodology/research-approach/deductive-approach-2/>. [Accessed: 18-Mar-2019].
 - [121] K. Wiegers, “First things first: prioritizing requirements,” *Softw. Dev.*, vol. 7, no. 9, pp. 48–53, 1999.
 - [122] M. Glinz, “On Non-Functional Requirements,” in *15th IEEE International Requirements Engineering Conference (RE 2007)*, 2007, pp. 21–26.
 - [123] International Organisation for Standardisation (ISO), “ISO 9000:2000 - Quality management systems -- Fundamentals and vocabulary,” *Standards Catalogue*, 2000. [Online]. Available: <https://www.iso.org/standard/29280.html>. [Accessed: 23-Feb-2019].
 - [124] F. Paetsch, A. Eberlein, and F. Maurer, “Requirements Engineering & Agile Software Development,” in *Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE’03)*, 2003, pp. 308–313.
 - [125] D. Zowghi and C. Coulin, “Requirements elicitation: A survey of techniques, approaches and tools,” *Eng. Manag. Softw. Requir.*, pp. 19–46, 2005.

- [126] V. Khatri and C. V. Brown, "Designing Data Governance," *Commun. ACM*, vol. 53, no. 1, p. 148, 2009.
- [127] P. Weill and J. W. Ross, *IT Governance: How Performers Manage IT Decision Rights for Superior Results*, 1st ed. Boston, MA: Harvard Business School Press, 2004.
- [128] J. Ladley, *Data Governance: How to Design, Deploy and Sustain an Effective Data Governance Program*, 1st ed. Waltham, MA: Morgan Kaufmann, 2012.
- [129] D. Plotkin, *In praise of Data Stewardship: An Actionable Guide to Effective Data Management and Data Governance*, 1st ed. Waltham, MA: Morgan Kaufmann, 2013.
- [130] Information Builders, "Seven Steps to Effective Data Governance for Healthcare Organizations: A White Paper," *White Pap.*, pp. 1–16, 2011.
- [131] M. Laanti, J. Similä, and P. Abrahamsson, "What Is Emphasized in Different Agile Definitions," *Eur. Conf. Softw. Process Improv.*, no. June, pp. 247–258, 2013.
- [132] J. Erikson, K. Lyytinen, and K. Siau, "Agile Modeling, Agile software Development, and extreme programming: The state of research," *J. Database Manag.*, vol. 16, no. 4, pp. 88–100, 2005.
- [133] A. Cockburn and J. Highsmith, "Agile Software Development: The People Factor," *Computer (Long. Beach. Calif.)*, vol. 34, no. 11, pp. 131–133, 2001.
- [134] J. Hunt, *Agile Software Construction*, 1st ed. London, UK: Springer, 2006.
- [135] K. Conboy and B. Fitzgerald, "Toward a conceptual framework for agile methods: a study of agility in different disciplines," in *Proceedings of XP/Agile Universe*, 2004.
- [136] Agile Manifesto, "Manifesto for Agile Software Development," *agilemanifesto.org*, 2001. [Online]. Available: <http://agilemanifesto.org/>. [Accessed: 21-Nov-2018].
- [137] B. Kotaiah and M. A. Khalil, "Approaches for development of Software Projects: Agile methodology," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 1, pp. 237–242, 2017.
- [138] D. Turk, R. France, and B. Rumpe, "Assumptions Under lying Agile Software Development Processes," *J. Database Manag.*, vol. 16, no. 4, pp. 62–87, 2005.
- [139] J. M. Fernandes and M. Almeida, "Classification and comparison of agile methods," *Proc. - 7th Int. Conf. Qual. Inf. Commun. Technol. QUATIC 2010*, pp. 391–396, 2010.
- [140] M. R. J. Qureshi and S. A. Hussain, "An adaptive software development process model," *Adv. Eng. Softw.*, vol. 39, pp. 654–658, 2008.
- [141] S. C. Misra, V. Kumar, and U. Kumar, "Identifying some important success factors in adopting agile software development practices," *J. Syst. Softw.*, vol. 82, no. 11, pp. 1869–1890, 2009.
- [142] T. Dybå and T. Dingsøy, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, no. 9–10, pp. 833–859, 2008.

- [143] T. Dingsøyr, S. Nerur, V. Balijepally, and N. Brede Moe, "A decade of agile methodologies: Towards explaining agile software development," *J. Syst. Softw.*, vol. 85, pp. 1213–1221, 2012.
- [144] A. F. Chowdhury and M. N. Huda, "Comparison between Adaptive Software Development and Feature Driven Development," *Proc. 2011 Int. Conf. Comput. Sci. Netw. Technol. ICCSNT 2011*, vol. 1, pp. 363–367, 2011.
- [145] M. Stoica, M. Mircea, and B. Ghilic-Micu, "Software Development: Agile vs. Traditional," *Inform. Econ.*, vol. 17, no. 4, pp. 64–76, 2013.
- [146] A. Faiza, A. Shabib, W. Usman, and M. Syed Shah, "Agile Software Development Models TDD, FDD, DSDM, and Crystal Methods: A Survey," *Int. J. Multidiscip. Sci. Eng.*, vol. 8, no. 2, 2017.
- [147] M. Rychly and P. Tichá, "A Tool for Supporting Feature-Driven Development," Brno, Czech Republic, 2008.
- [148] S. R. Palmer and M. Felsing, "A Practical Guide to Feature-Driven Development," in *The Coad Series*, P. Coad, Ed. Upper Saddle River, New Jersey, NJ: Prentice Hall PTR, 2002.
- [149] M. L. Despa, "Comparative study on software development methodologies," *Database Syst. J.*, vol. 5, no. 3, pp. 37–56, 2014.
- [150] T. J. Gandomani, H. Zulzalil, A. A. A. Ghani, A. B. M. Sultan, and M. Z. Nafchi, "Obstacles in moving to agile software development methods; At a Glance," *J. Comput. Sci.*, vol. 9, no. 5, pp. 620–625, 2013.
- [151] J. A. Livermore, "Factors that significantly impact the implementation of an agile software development methodology," *J. Softw.*, vol. 3, no. 4, pp. 31–36, 2008.
- [152] E. Spertus and L. A. Stein, "Just-In-Time Databases and the World-Wide Web," in *CIKM '98*, 1998.
- [153] K. S. Ahmad, N. Ahmad, H. Tahir, and S. Khan, "Fuzzy-MoSCoW: A fuzzy based MoSCoW method for the prioritization of software requirements," *2017 Int. Conf. Intell. Comput. Instrum. Control Technol. ICICICT 2017*, vol. January, pp. 433–437, 2018.
- [154] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. R. Mahrin, "A systematic literature review of software requirements prioritization research," *Inf. Softw. Technol.*, vol. 56, no. 6, pp. 568–585, 2014.
- [155] J. Tidwell, *Designing Interfaces*, 1st ed. Sebastopol, CA: O'Reilly Media, Inc., 2005.
- [156] P. T. A. Junior and L. V. L. Filgueiras, "User modeling with personas," in *Proceedings of CLICH'05*, 2006, pp. 277–282.
- [157] J. Pruitt and J. Grudin, "Personas : Practice and Theory," *Proc. Conf. Des. user Exp.*, pp. 1–15, 2003.

- [158] Z. Koltay and T. Kornelia, "Personas and User-centred Visioning Process," in *Proceedings of the 2008 Library Assessment Conference: building effective, sustainable, practical assessment.*, 2008, pp. 187–194.
- [159] R. Baker, *Agile UX Storytelling: Crafting Stories for Better Software Development*, 1st ed. Plano, TX: CA Press, 2017.
- [160] H. Cobbold, "Prototype Application Design: FACE Companion," *User Experience Design*. UWTSD, Swansea, UK, 2019.
- [161] R. M. Chopade and N. S. Dhavase, "Agile Software Development: Positive and Negative User Stories," in *2017 2nd International Conference for Convergence in Technology (I2CT)*, 2017.
- [162] T. Neil, *Mobile Design Pattern Gallery: UI Patterns for iOS, Android and More*, 1st ed. Sebastopol, CA: O'Reilly Media, Inc., 2012.
- [163] Apple Inc., "Split Views - Views - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/views/split-views/>. [Accessed: 27-Mar-2019].
- [164] N. Babich, "Mobile UX Design: List View and Grid View," *UX Planet: Medium*, 2016. [Online]. Available: <https://uxplanet.org/mobile-ux-design-list-view-and-grid-view-8f129b56fd5b>. [Accessed: 27-Mar-2019].
- [165] Apple Inc., "Pickers - Controls - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/controls/pickers/>. [Accessed: 27-Mar-2019].
- [166] Apple Inc., "Switches - Controls - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/controls/switches/>. [Accessed: 27-Mar-2019].
- [167] Apple Inc., "Tables - Views - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/views/tables/>. [Accessed: 27-Mar-2019].
- [168] Apple Inc., "Search Bars - Bars - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/bars/search-bars/>. [Accessed: 27-Mar-2019].
- [169] M. Fowler and K. Scott, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 2nd ed. Reading, MA: Addison Wesley, 1999.
- [170] P. Coad, E. Lefebvre, and J. De Luca, *Java Modeling in Color with UML: Enterprise Components and Process*, 1st ed. Upper Saddle River, NJ: Prentice Hall PTR, 1999.
- [171] S. R. Palmer, "Just Typical: UML Stereotypes and Class Archetypes | Extending UML with

- Stereotypes,” *Programming*, 2002. [Online]. Available: <http://www.informit.com/articles/article.aspx?p=29454>. [Accessed: 21-Mar-2019].
- [172] S. W. Ambler, “UML 2 Class Diagram Guidelines,” *Agile Modeling*, 2018. [Online]. Available: <http://agilemodeling.com/style/classDiagram.htm>. [Accessed: 21-Mar-2019].
- [173] S. W. Ambler, “Agile Model Driven Development (AMDD): The Key to Scaling Agile Software Development,” *Agile Model Driven Development (AMDD)*. [Online]. Available: <http://www.agilemodeling.com/essays/amdd.htm>. [Accessed: 12-Mar-2019].
- [174] D. J. Anderson, “Feature-Driven Development: Towards a TOC, Lean and Six Sigma solution for software engineering,” *TOC ICO World Conf. Oct. 2004*, no. October, 2004.
- [175] S. Goyal, “Major Seminar On Feature Driven Development: Techniques for Project Management and Software Engineering,” Technical University Munich, 2007.
- [176] M. Blaha, “Referential Integrity Is Important For Databases,” *Model. Consult. Corp.*, pp. 1–3, 2005.
- [177] MSDN, “SQL Server Audit (Database Engine),” *Auditing*, 2016. [Online]. Available: <https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-database-engine?view=sql-server-2017>. [Accessed: 26-Mar-2019].
- [178] J. Worthen, “Creating A Log Table To Track Changes To Database Objects In SQL Server,” *SQL Server*, 2018. [Online]. Available: <https://jackworthen.com/2018/03/19/creating-a-log-table-to-track-changes-to-database-objects-in-sql-server/>. [Accessed: 15-Mar-2019].
- [179] R. Drizin, “Audit Tables and Data Versioning on SQL Server,” *Articles*, 2016. [Online]. Available: <https://www.codeproject.com/Articles/1112660/Audit-Tables-and-Data-Versioning-on-SQL-Server>. [Accessed: 26-Mar-2019].
- [180] MSDN, “Temporal Tables,” *Tables*, 2016. [Online]. Available: <https://docs.microsoft.com/en-us/sql/relational-databases/tables/temporal-tables?view=sql-server-2017>. [Accessed: 26-Mar-2019].
- [181] C. M. Barnum, *Usability Testing Essentials: Ready, Set... Test!*, 1st ed. Burlington, MA: Morgan Kaufmann, 2010.
- [182] E. Wong, “Heuristic Evaluation: How to Conduct a Heuristic Evaluation,” *Interaction Design Foundation: Literature*, 2019. [Online]. Available: https://www.interaction-design.org/literature/article/heuristic-evaluation-how-to-conduct-a-heuristic-evaluation?utm_source=facebook&utm_medium=sm. [Accessed: 11-Apr-2019].
- [183] N. E. Jacobsen and B. E. John, “Two Case Studies in Using Cognitive Walkthrough for Interface Evaluation,” in *CMU-CS-00-132*, 2000.
- [184] M. H. Blackmon, *Cognitive Walkthrough Cognitive Walkthrough Methodology*, vol. 1. Great

- Barrington, MA: Berkshire Publishing Group, 2004.
- [185] Apple Inc., "Segmented Controls - Controls - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/controls/segmented-controls/>. [Accessed: 27-Mar-2019].
 - [186] J. Brooke, "SUS - A quick and dirty usability scale," *Smart Phone Appl. People with Brain Inj.*, vol. Smart phon, pp. 1–8, 2011.
 - [187] S. Krug, *Rocket surgery made easy: The do-it-yourself guide to finding and fixing usability problems*, 1st ed. Berkeley, CA: New Riders, 2010.
 - [188] J. McGinn and A. R. Chang, "RITE+Krug: A Combination of Usability Test Methods for Agile Design," *J. Usability Stud.*, vol. 8, no. 3, pp. 61–68, 2013.
 - [189] Usability.gov, "System Usability Scale (SUS)," 06-Sep-2013. [Online]. Available: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>. [Accessed: 15-Apr-2019].
 - [190] J. R. Lewis, "Usability Testing," in *Handbook of Human Factors and Ergonomics*, 3rd ed., Hoboken, NJ: John Wiley, 2009, pp. 1275–1316.
 - [191] S. Krug, *Don't make me think, revisited: a common sense approach to Web usability*, 3rd ed. San Francisco, CA: New Riders, 2014.
 - [192] J. Sauro, "Does Prior Experience Affect Perceptions Of Usability?," *Prior Exposure*, 2011. [Online]. Available: <https://measuringu.com/prior-exposure/>. [Accessed: 18-Apr-2019].
 - [193] L. Lindstrom and R. Jeffries, "Extreme Programming and Agile Software Development Methodologies," *Inf. Syst. Manag.* 21, vol. 3, pp. 41–53, 2004.
 - [194] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile Software Development Methods: Review and Analysis," *VTT Publ.* 478, 2002.
 - [195] A. Hamdan Mohammad, T. Alwada'n, and J. Ababneh, "Agile Software Methodologies: Strength and Weakness," *Int. J. Eng. Sci. Technol.*, vol. 5, no. 3, 2013.
 - [196] J. Gulliksen, B. Göransson, I. Boivie, S. Blomkvist, J. Persson, and Å. Cajander, "Key principles for user-centred systems design," *Behav. Inf. Technol.*, vol. 22, no. 6, pp. 397–409.
 - [197] S. Chamberlain, H. Sharp, and N. Maiden, "Towards a Framework for Integrating Agile Development and User-Centred Design," pp. 143–153, 2006.
 - [198] M. Steen, "Tensions in human-centred design," *CoDesign*, vol. 7, no. 1, pp. 45–60, 2011.
 - [199] P. Kumar, J. Kinch, and S. O. Ogunlana, "Management of risks in information technology projects: a case study," *Ind. Manag. Data Syst.*, vol. 107, no. 2, pp. 286–295, 2004.
 - [200] B. Sharif, S. A. Khan, and M. Wasim Bhatti, "Measuring the Impact of Changing Requirements

- on Software Project Cost: An Empirical Investigation,” *Int. J. Comput. Sci. Issues*, vol. 9, no. 3, pp. 170–174, 2012.
- [201] N. Nurmaliani, D. Zowghi, and S. P. Williams, “Requirements Volatility and Its Impact on Change Effort: Evidence-based Research in Software Development Projects Return to Published Papers,” Adelaide, Australia, 2006.
- [202] N. Merchant, “Continuous Delivery and Kanban: The Perfect Marriage,” *Continuous Delivery and Kanban*, 2016. [Online]. Available: <https://itglue.com/blog/continuous-delivery-and-kanban-the-perfect-marriage/>. [Accessed: 08-Oct-2018].
- [203] S. Rose, “GDPR Compliancy with OneDrive and SharePoint - Microsoft Tech Community - 191126,” *Microsoft OneDrive Blog*, 2018. [Online]. Available: <https://techcommunity.microsoft.com/t5/Microsoft-OneDrive-Blog/GDPR-Compliancy-with-OneDrive-and-SharePoint/ba-p/191126>. [Accessed: 15-Mar-2019].
- [204] Microsoft Inc., “Journey to GDPR Compliance.”
- [205] Apple Inc., “Privacy Policy,” *Legal*, 2018. [Online]. Available: <https://www.apple.com/legal/privacy/en-ww/>. [Accessed: 15-Mar-2019].
- [206] Apple Inc., “We give developers powerful tools to protect your data,” *Our Approach to Privacy*. [Online]. Available: <https://www.apple.com/privacy/approach-to-privacy/>. [Accessed: 15-Mar-2019].
- [207] Microsoft Inc., “Microsoft Azure: Overview, Security, GDPR,” *Trust Centre*, 2019. [Online]. Available: <https://www.microsoft.com/en-us/TrustCenter/CloudServices/Azure/GDPR>. [Accessed: 15-Mar-2019].
- [208] University of Wales Trinity Saint David and M. Briggs, “Research Data Management Policy,” Carmarthen, Swansea, Lampeter, London, 2016.
- [209] M&DC, “Kanban,” *Management and Development Centre*. [Online]. Available: [http://www.mdcegypt.com/pages/management approaches/Lean enterprise/Kanban/Kanban.asp](http://www.mdcegypt.com/pages/management%20approaches/Lean%20enterprise/Kanban/Kanban.asp). [Accessed: 11-Oct-2018].
- [210] A. D. Mimits, “JIT and Lean Production,” *Module II: Production and Operations Management*, 2015. [Online]. Available: <https://slideplayer.com/slide/7324757/>. [Accessed: 11-Oct-2018].
- [211] LeanKit, “10 Kanban Board Examples,” *Kanban Learn*, 2018. [Online]. Available: <https://leankit.com/learn/kanban/kanban-board-examples-for-development-and-operations/>. [Accessed: 12-Dec-2018].

9. BIBLIOGRAPHY AND ADDITIONAL WORK(S)

A. G. Taylor, *SQL for Dummies*, 5th ed. Hoboken, NJ: Wiley Publishing Inc., 2013.

A. Harrison and R. van Hoek, *Logistics Management and Strategy: Competing through the supply chain*, 3rd ed. Harlow, Essex: Pearson Education, 2008.

Apple Inc., "Pickers - Controls - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/controls/pickers/>. [Accessed: 27-Mar-2019].

Apple Inc., "Search Bars - Bars - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/bars/search-bars/>. [Accessed: 27-Mar-2019].

Apple Inc., "Segmented Controls - Controls - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/controls/segmented-controls/>. [Accessed: 27-Mar-2019].

Apple Inc., "Split Views - Views - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/views/split-views/>. [Accessed: 27-Mar-2019].

Apple Inc., "Switches - Controls - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/controls/switches/>. [Accessed: 27-Mar-2019].

Apple Inc., "Tables - Views - iOS - Human Interface Guidelines - Apple Developer," *Apple Developer*, 2019. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/ios/views/tables/>. [Accessed: 27-Mar-2019].

B. Srinidhi and G. K. Tayi, "Just in time or just in case? An explanatory model with informational and incentive effects," *J. Manuf. Technol. Manag.*, vol. 15, no. 7, pp. 567–574, 2004.

B. Unhelkar, *The Art of Agile: A Composite Approach for Projects and Organisations*, 1st ed. Boca Raton, FL: CRC Press, 2013.

C. M. Harland, "Supply Chain Management," *Br. J. Manag.*, vol. 7, no. Special, pp. 63–80, 1996.

D. Andreea Căilean and K. Sharifi, "Mobile ERP: A literature review on the concept of Mobile ERP systems," Jonkoping University, 2013.

- D. Stindt, "A generic planning approach for sustainable supply chain management - How to integrate concepts and methods to address the issues of sustainability?," *J. Clean. Prod.*, vol. 153, pp. 146–163, 2017.
- E. Bendoly and T. Schoenherr, "ERP system and implementation-process benefits: Implications for B2B e-procurement," *Int. J. Oper. Prod. Manag.*, vol. 25, no. 3/4, pp. 304–319, 2005.
- F. Febrero, C. Calero, and M. Á. Moraga, "Software reliability modeling based on ISO/IEC SQuaRE," *Inf. Softw. Technol.*, vol. 70, pp. 18–29, 2016.
- H. Reefke and D. Sundaram, "Key themes and research opportunities in sustainable supply chain management – identification and evaluation," *Omega (United Kingdom)*, vol. 66, no. B, pp. 195–211, 2017.
- J. Gebauer, M. J. Shaw, and K. Zhao, "The Efficacy of Mobile e-Procurement: A Pilot Study," in *Proceedings of the 36th Hawaii International Conference on System Sciences*, 2003, pp. 1–10.
- J. Haugland, "Towards Lean Procurement: Challenges and Opportunities," Thesis, University of Stavanger, 2015.
- J. Mont, "The Risks and Benefits of Employee-Owned Devices," *Compliance Week*, vol. 9, no. 103, pp. 48–49, 2012.
- J. Teeravaraprug, K. Kitiwanwong, and N. Saetong, "Relationship model and supporting activities of JIT, TQM and TPM," *Songklanakarin J. Sci. Technol.*, vol. 33, no. 1, pp. 101–106, 2011.
- L. Beighley, *Head First SQL*. Sebastapol, CA: O'Reilly, 2007.
- M. Ellis-Lee, "Let's Get Physical: When Skeuomorphism and Flat Design Meet," *Flywheel Strategic: Thinking [Image]*. [Online]. Available: <https://www.flywheelstrategic.com/thinking/post/flywheel-blog/2017/10/27/let-s-get-physical-when-skeuomorphism-and-flat-design-meet>. [Accessed: 27-Mar-2019].
- O. Gordieiev, V. Kharchenko, N. Fominykh, and V. Skylar, "Evolution of Software Quality Models in Context of the Standard ISO 25010," *Adv. Intell. Syst. Comput.*, vol. 286, no. September 2016, 2014.
- R. V Ramasesh, "Recasting the Traditional Inventory Model to Implement Just-in-Time Purchasing," *Prod. Invent. Manag. Journal; First Quart.*, vol. 31, no. 1, pp. 71–75, 1990.
- Rail Safety and Standards Board (RSSB), "Railway Vehicle Maintenance," *Railw. Gr. Stand. GM/RT2005*, no. 5, p. 21, 2012.
- S. McLellan, A. Muddimer, and S. C. Peres, "The Effect of Experience on System Usability Scale Ratings," *Int. J. Usability Stud.*, vol. 7, no. November 2011, pp. 56–67, 2012.

S. Stern, A. Behrendt, E. Eisenschmidt, S. Reimig, L. Schirmers, and I. Schwerdt, "The Rail Sector's Changing Maintenance Game," 2017.

Wisegate, "Sample Corporate Mobile Device Acceptable Use and Security Policy," *webpage*, 2017. [Online]. Available: http://wisegateit.com/resources/downloads/wisegate-sample-byod-policy.pdf?_ga=1.166862838.993227471.1475359178. [Accessed: 08-May-2018].

10. APPENDICES

Section 10 documents any material referenced throughout the above sections whilst providing additional, supplementary material that was not considered integral to the main report body. Throughout the following sub-sections, additional background information on Agile development methodologies, a proposed data governance policy, private communications received from GWR, SQL database code and the various testing and formal reports for the wireframe and prototype application.

10.1 Traits of Agile Methodologies

The theories of agility and diverse team skillsets contrast with traditional Software Development Life Cycle (SDLC) structures, processes and documentation that creates rigidity, but little project adaptability [133],[69],[70],[74] to the work of Conboy and Fitzgerald [135].

“Individuals and interactions over processes and tools” [194]. Agile structures focus closely on employee collaboration. This team-centric approach, known as the team communication principle, is typically achieved with small, tight development groups that produce quality products to a tight time schedule [138],[145],[194]. This is a key element of Agile methodologies as it shows the consideration of waste and wasteful production found in typical software development methodologies, comparable to views held by Ohno regarding wasteful manufacturing practice [133]. Cockburn and Highsmith [133], suggesting that *“people trump process[es]”* [133], propose a physically close team structure and the use of quick and practical design tools, such as white or Kanban boards that can readily accommodate project change.

“Working software over comprehensive documentation” [194]. The visibility assumption principle denotes the importance of working prototypes or design simulations during the development stage, helping to define project progress. Extensive technical documentation on the other hand can be hard to quantify, especially for clients, and struggles to represent tangible progress [138],[194]. Abrahamsson [194] suggests that the significance of code documentation is related to the scope of the implementation, developed by Turk, [138] who notes how enterprise-level systems demand precise documentation to ensure business rules are precisely fulfilled.

“Customer collaboration over contract negotiation” [194]. Dingsøyr et. al [143] suggest that this, combined with short development iterations noted above increases creativity and productivity, indirectly providing a system capable of offering design and development fluidity. Large benefits can be seen in human-computer interaction (HCI) as interface design decisions can be shared as they arise, the success of which however relies on appropriate customer interaction [138]. Whilst client contact is considered one of the biggest benefits of this family of methodologies, the busy nature of firms can signify inadequate contact and insufficient feedback during the critical construction iterations [195]. The significance of the user has been further explored by the User Centred Design (UCD) method, and underlying principles. Sitting relatively adjacent to Agile, its design and development approach is also cyclical. Equally focused on continual modification and feedback in a team environment, the client sits right at the heart to *“dominate the design of the interface”* [196], thus influencing *“the design of the rest of the system”* [196],[197],[198].

“Responding to change over following a plan” [194] is a key part of agile. Methodologies allow change thanks to the cyclical nature of the design, development and testing activities – typically undertaken by software developers and the client. Adjustments highlighted during a development iteration therefore should be both easily accommodated during product development and open to the customer to demand [138],[145],[194]. Mohammad, Alwada’n and Ababneh [195] however insinuate that the lack of long term vision creates the issue of *“refactoring”*, changes that occur during feature integration with knock-on effects on documentation.

Conboy and Fitzgerald [135], building on the idea of flexibility, present a solution that shows the significance of agility in the software development field. **Figure 69** shows an exploratory framework that places the financial outlay of an activity, the y axis, against the development methodologies ability to identify and handle project change, represented by the x axis [135]. The process of ‘creation’, whereby the development team themselves initiate project change, can support considerable change at a relatively low cost thanks to low stakeholder or management intervention. However, activities relating to predicting change (‘proaction’) and error correction (‘reaction’) are substantially more demanding [135]. This graph represents the widely understood premise that the later changes are made in a project the greater the financial and time outlay required for their rectification [135],[199],[200]. The term ‘agility’ sits between ‘proaction’ and ‘reaction’ to identify the point where actively searching for program code errors (and other changes associated with natural project progression, often referred to as *“requirements volatility”* [201]) takes place, ultimately aiming to reduce the impact of unforeseen project changes [135].

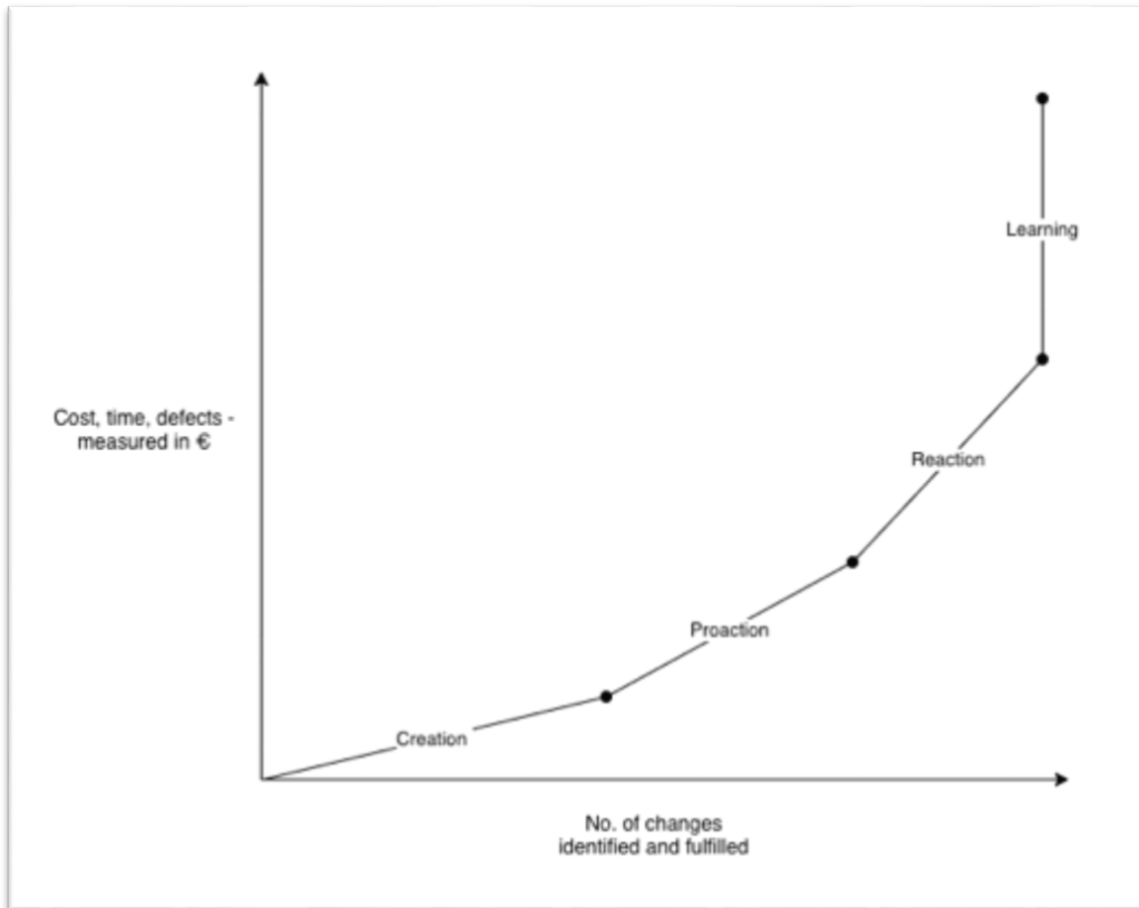


Figure 69: Cost and change accommodation of software development activities [63]

The process of 'learning' from past development experiences is hard to directly measure as its outcomes, the changes owing to this process, are hard to explicitly identify [135]. Conboy and Fitzgerald [135] suggest that the predicted changes refer to the continuous improvement philosophy and changes that might take effect during forthcoming projects. A visual representation of the impact of the learning process is offered below in **Figure 70**.

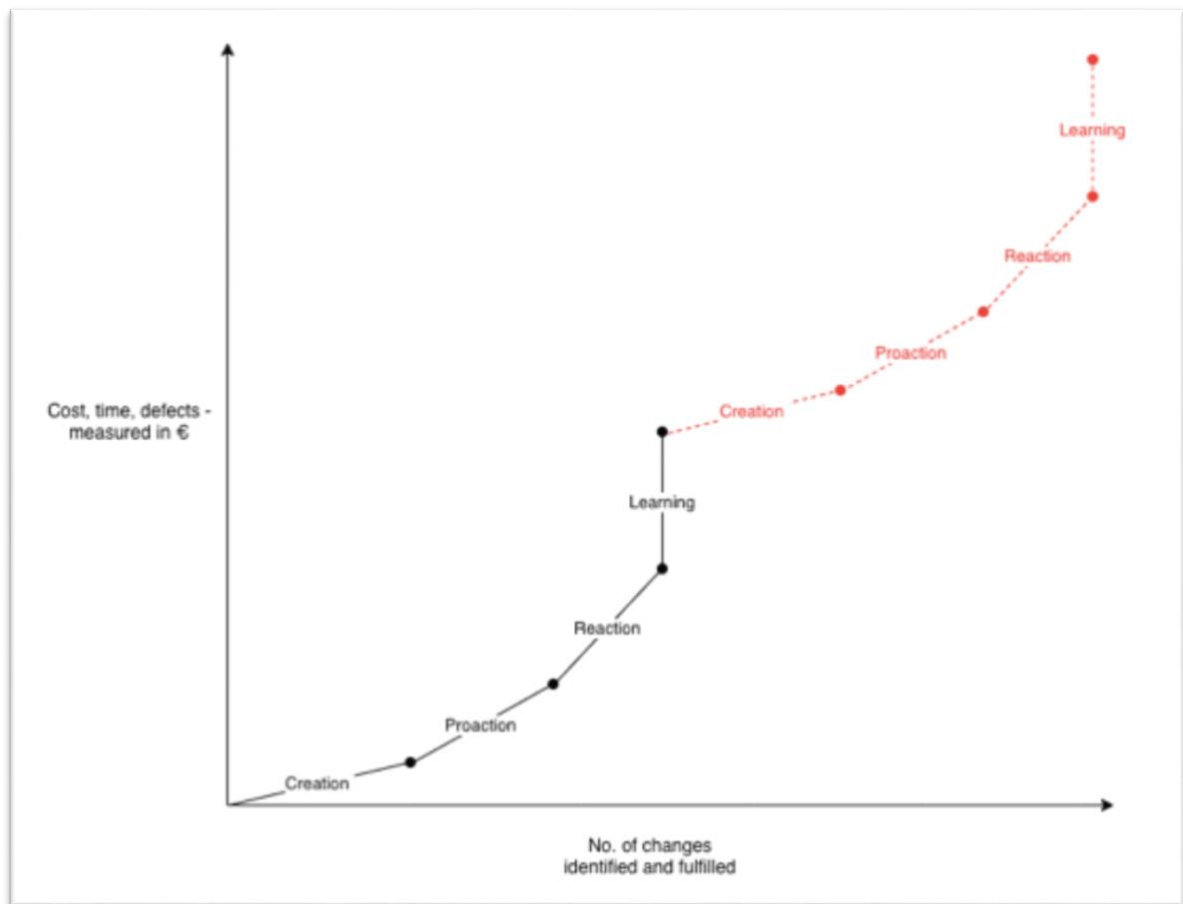


Figure 70: The possible long-term effect of 'Learning' on future projects [63]

Given the shared traits of agility and precise task management, several tools with foundations in manufacturing have also been applied to the field of software development and IT project management. Understanding the benefits of Kanban in continuous delivery configurations [202],[46], Anderson's adaptation for the software development environment closely monitors WIP levels and visualises development workflows. Practically Anderson's board concentrates on a team focus, reducing wasteful practice (typically part or fully complete stockpiled software solutions) and helps shorten lead times [42],[202], making this system clearly aligned with Ohno's original conception and deployment of his production cards. This Kanban board is a commonly used tool to both view and restrict work-in-progress, as shown in **Figure 72** [46].

The board states that processes sat at various stages throughout the development cycle can only progress when there is the capacity at the subsequent process to contain the given object or state. The tasks sat in "Waiting" can only be accepted at the first stage ("In Design") when suitable capacity here exists (i.e. a task currently occupying "In Design" must be pulled through by "In Progress" to create this), consequently allowing the cascading of jobs. This movement however is

reliant on job capacity fashioned further along the software development cycle, particularly with capacity afforded by the “In Progress” phase [46].

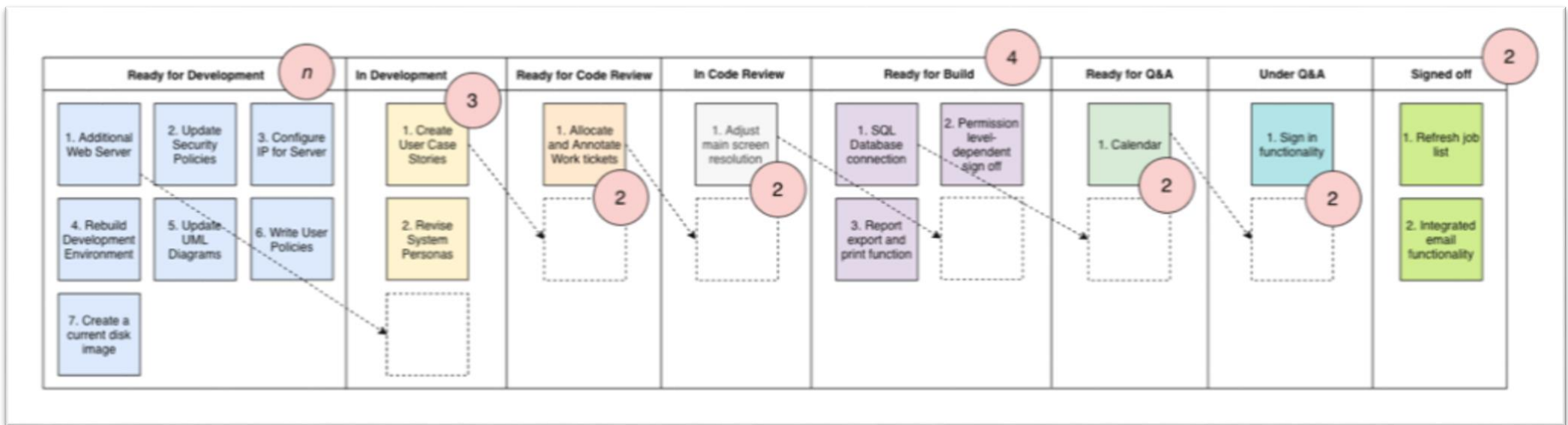


Figure 71: A Kanban board designed to allocate, balance and limit software project tasks in progress [211]

10.2 Data Governance Policy

10.2.1 Policy Purpose

Table 17: Data Governance Framework Overview

Version No.	Approved by	Approved date	Effective date	Next Review
1.1	Dr. Kemi Ademoye & Dr. Nik Whitehead	05/10/18	05/10/18	<i>n/a</i>
Policy Overview statement				
Purpose	The purpose of the following Data Governance Policy is to: • Define the roles and responsibilities for different data creation and usage types, cases and/or situations, and to establish clear lines of accountability. • Develop best practices for effective data management and protection during the project lifespan. • Protect GWR's maintenance data against all project-related external threats. • Ensure that the project individual complies with all applicable UWTSD laws, regulations and ethical standards. • Ensure that the data trail is appropriately documented throughout the project so as to ensure transparency regarding access, usage and/or modification within the confines of the project, storage mechanisms and final documentation and/or presentation.			
Scope	This policy applies to all data obtained for the purpose of this project from GWR. This policy covers, but is not limited to, GWR's maintenance data in any form, including print, electronic, audio visual, backup and archived data. This policy applies to the project individual and the project supervisors.			

10.2.2 Governance and Ownership

Table 18: Policy Governance and Ownership

Governance and Ownership	
<i>Data Governance Role</i>	<i>Data Governance Responsibility</i>
Data Custodian	GWR as a firm are the custodians of the data and any information derived from this data
Chief Data Officer	GWR's Chief Data Officer is responsible for the overall management of the of firm's data, and tasks associated with the role of Data Governance
Data Owner	Data used as part of this project has been provided by <i>Colin Page</i> and <i>Chris Lunn</i> as they are responsible for ensuring suitable protocols are in place to guide the appropriate use of the firm's data assets locally (on site).
Data Stewards	<i>Dr. Nicola Whitehead</i> and <i>Dr. Kemi Ademoye</i> are data stewards who both approve access under delegation from the Data Owner based on the User's role and intended use of this data.
Data Creators	The Data Creator is both the creator of the data in its original format, i.e. the maintenance schedule documents and those responsible for the quality and integrity, implementation and enforcement of data management within a research project, the project individual, <i>Henry Cobbold</i> .
Data Specialists	Several key lecturing staff, both those directly involved with supporting the project and the individual and subject matter experts in relation to data or information assets. Those concerns are as follows: • <i>Dr. Nicola Whitehead</i>

10.2.3 Data Use

Table 19: Data Use Policy

Data Use	
Access	Access to electronic and/or print versions of data has been provided by <i>Colin Page</i> and <i>Chris Lunn</i> of GWR with permission, where appropriate, from other company employees and/or representatives.
Storage	Physical data is being kept by the lead developer at all times during the day and when not required on site in University is kept in the individual's locker on-campus. During out of hours periods it is being kept at the term residence. Electronic copies of data are being kept on the individual's work Laptop and Desktop through both their machine and the University OneDrive account, the service of which is covered by Microsoft's Data Use and Privacy Policies [203],[204], and the storage of which is within the UK and subject to domestic legislation. The storage on both Laptop and Desktop is subject to Apples Privacy Policy [205],[206]. The creation of new content, based on electronic copies, is hosted on Microsoft's Azure platform and subject to corresponding regulation [207].
Manipulation and/or Creation	Data is being created and/or manipulated for the purposes of the project as deemed suitable by the individual for the purpose of academic study.
Display and Presentation	For the purpose of academic accreditation, the data in modified and/or original format has been presented and or/distributed to both project supervisors as deemed necessary by the project student. Lecturers and/or all members of academic staff who have received and/or distributed said data within the University comply with UWTSD's comprehensive Data Governance Policy [208]. No presentation and/or distribution has occurred to individuals and/or organisations outside the scope of the university with the exception of secure storage purposes. Both Apple Inc. and Microsoft Inc. are subject to suitable data protection regulation.
Deletion	On completion of the project, its marking and all academic verification, access to data provided by the Data Owner will be terminated with its timely and appropriate destruction.

10.2.4 Declaration

Table 20: Data Governance Framework Declaration

Declaration and Signature of Consent	
Colin Page / Chris Lunn	<i>Signature here</i>
Henry Cobbold	<i>Signature here</i>
Dr. Kemi Ademoye & Dr. Nicola Whitehead	<i>Signature here</i>
Revision History	
<i>n/a</i>	

It is important to note that no personal details have been used in the project beyond several cases of employee names present in some data obtained. These have been anonymised.

10.3 GWR Private Communications

10.3.1.1 Scheduled Maintenance Plan Sheet

[illegible]

10.3.2 Short Term Plan (STP) Sheet: Power cars

LAIRA SHORT TERM PLAN 10/07/18

POWER CARS

Changes ***	Veh. No	Exam	Expected Due Date	Req'd at Laira	Vice	Notes
	43097	C + Group checks	11/07	11/07	43091	
	43040	C + Data Alchemist Logger fault	13/07	12/07	43139	DEM to attend
	43053	Van door reps + ATP checks	-	13/07	43150	Currently LOOU DEM to attend
	43069	C + UAT	13/07	14/07	43153	Mileage Saving Req'd
	43130	T/T	-	14/07	N/A	Arriving with OC32, C4D set
	43185	D	-	15/07	43053	D Exam less C Exam items Kev Edwards to advise
	43094	C + UAT + Data Alchemist Logger Fitment	16/07	15/07	43078	M.Gunn to attend
	43009	C	17/07	17/07	43185	
	43129	C + UAT + Data Alchemist Logger Fitment	19/07	18/07	43040	M.Gunn to attend
	43189	D	28/07	19/07	43041	

43171 + 43097 for LA73 (available for service 17/07) Reduce to 9 P/Cars

SHUNTERS + BARRIERS

Changes ***	Veh. No	Exam	Expected Due Date	Req'd at Laira	Vice	Notes
	08644	A	19/07	-	-	
	08641	A	29/07	-	-	

10.3.3 Short Term Plan (STP) Sheet: Trailer Vehicles

TRAILER CAR SETS

Changes ***	Set No	Exam	Time at Laira	Expected Due Date	Req'd at Laira	Vice	Notes
	OC31	E + UAT	7 days	23/07	12/07	OC41	
	LA61	D + UAT	4 days	19/07	17/07	OC36	
	OC32	C4D	14 days	-	14/07	N/A	No Replacement 43130 for T/T Not 40757 Bogies due 17 - 20/07
	LA08	D exam	4 days		21/07	LA61	Pending D + Delap
	OC35	C	3 days	20/07	17/07	OC31	
	OC47	C	3 days	23/07	25/07	LA08	
	LA73				28/07	OC47	
	OC37	C		25/07		LA73	

Changes ***	Set No	Exam	Time at Laira	Expected Due Date	Req'd at Laira	Vice	Notes
	OC36	E	5 days	17/07	10/07	OC44	14 Month Rule 44003 A/C reps
	LA08	D + Delap	4 days	18/07	12/07	OC41	42263 A/C hard wired
	OC32	C4D	14 days	-	14/07	N/A	No Replacement 43130 for T/T Not 40757 Bogies due 17 - 20/07
	LA61	D + UAT	4 days	19/07	15/07	OC36	
	OC31	E + UAT	5 days	19/07	16/07	LA08	
***	LA16	Delap	TBA	-	17/07	N/A	To be the 'stand down' set
	OC35	C	3 days	20/07	19/07	LA61	

10.3.4 Initial Status Sheet (ISS) Form:

TRAILER CAR SET INITIAL STATUS SHEET

SET NUMBER: OC36

EXAM DUE: E EXAM

START DATE: 11/07/18

VEH NO.	PIR DEFECTS	OUTSTANDING MATERIALS	TREAD DAMAGE TO CHECK OR T/T REQ.	ROOF/TANK/FAN REPS	AUTO DOOR MOD DUE?	BRAKE COCK SEALS?	A/C FAULTS?	DELAP & BODY REPS ON R2?	DISC FRACTURES
41110									
40809									
42346				NO2 TOILET FAN NOISY		Y			
42517							A/C NOT WORKING		
WAITS 42349						Y			
WAITS 42138						Y			
42089									
44003				WATER LEAK FROM ROOF NO1 END			A/C MODULE TO CHANGE SUSPECT LOW GAS		

OTHER SIGNIFICANT DEFECTS:

AIR CON REPORTED POOR ON SET
 42346 SPLIT AIR PIPE NO2 VESTIBULE
 40809 MICROWAVE NOT WORKING
 42517 DOOR 4 CDL LIGHT DIM & FLICKERING
 40809 BLODGETT OVEN NOT WORKING
 40809 1 X HOB NOT WORKING
 42349 NO1 AUTO DOOR U/S

10.3.5 WCG Reprint Costings

WORK CONTENT GUIDE COSTING

185 Work Content Guides (approximately 20000 sheets of paper in total) Due to wear and tear require renewal at least once per year.

PRINTING: 20000 sheets at 45 seconds per sheet (to print 2-sided) = 250 hours at £20 per hour = £5,000

Each page then needs to be punched, plastic protective covers added and then all joined together using a binding comb

ASSEMBLY: 185 guides at 30 mins average per guide to assemble = 92.5 hours at £20 per hour = £1,850

Costs for materials assuming standard price of 10p per A4 sheet printed in colour double sided, plastic covers at £12.49/100 and binding combs £36

MATERIALS: 20000 sheets at £0.10 per sheet = £2,000.00

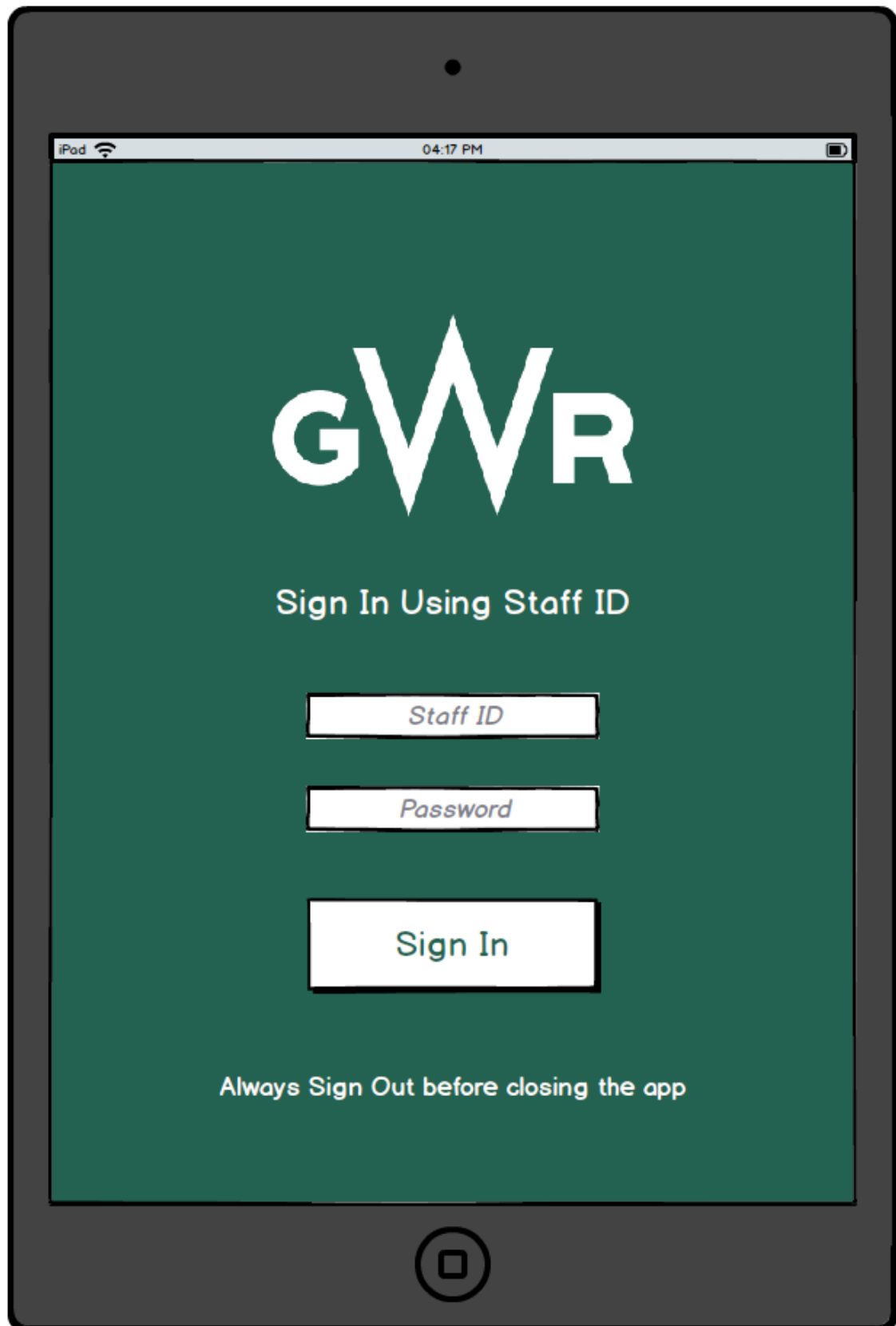
185 combs at £0.74 per comb = £136.90

370 covers at £0.125 per cover = £46.25

ANNUAL TOTAL : £9,033.15

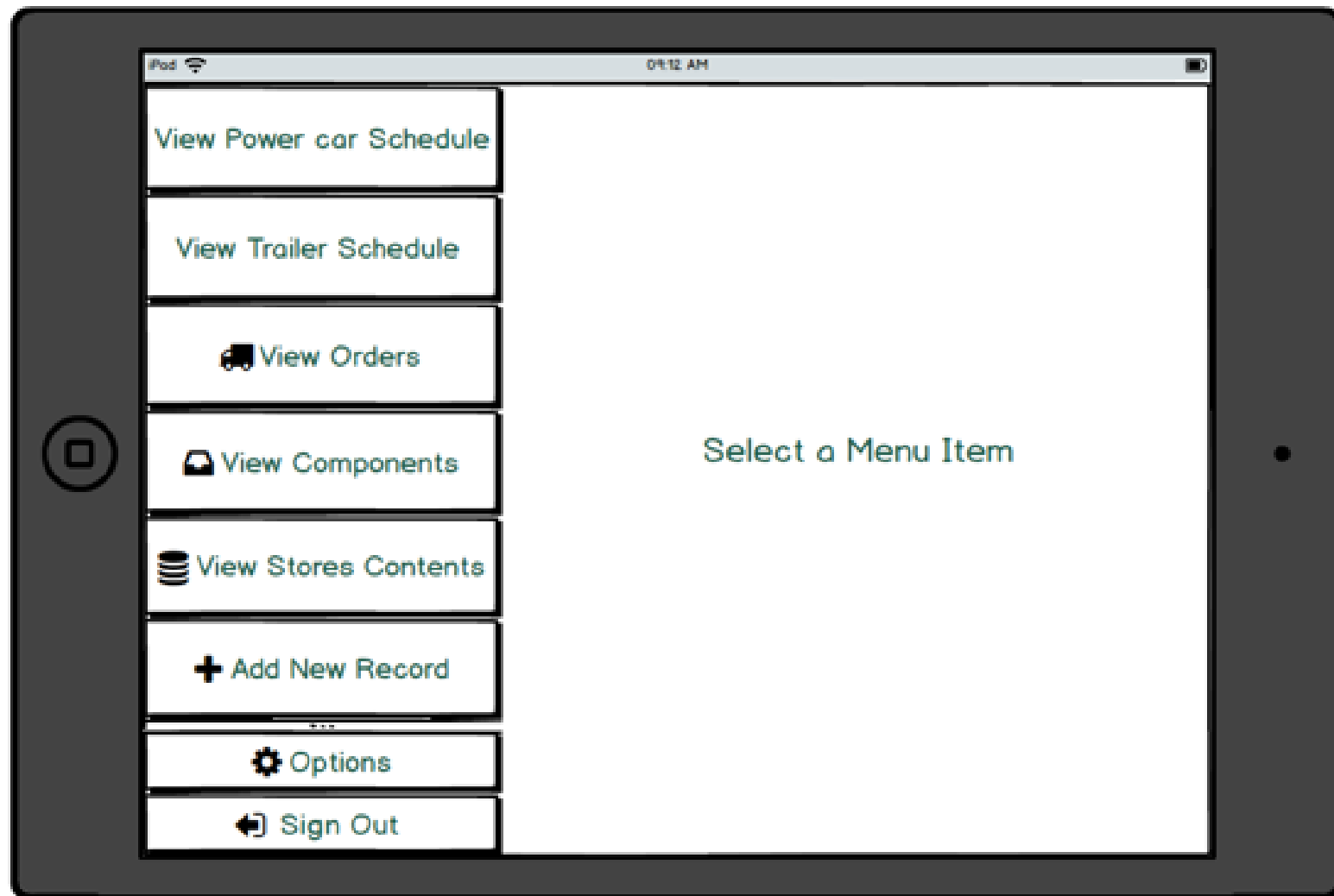
10.4 Interface Wireframe Designs

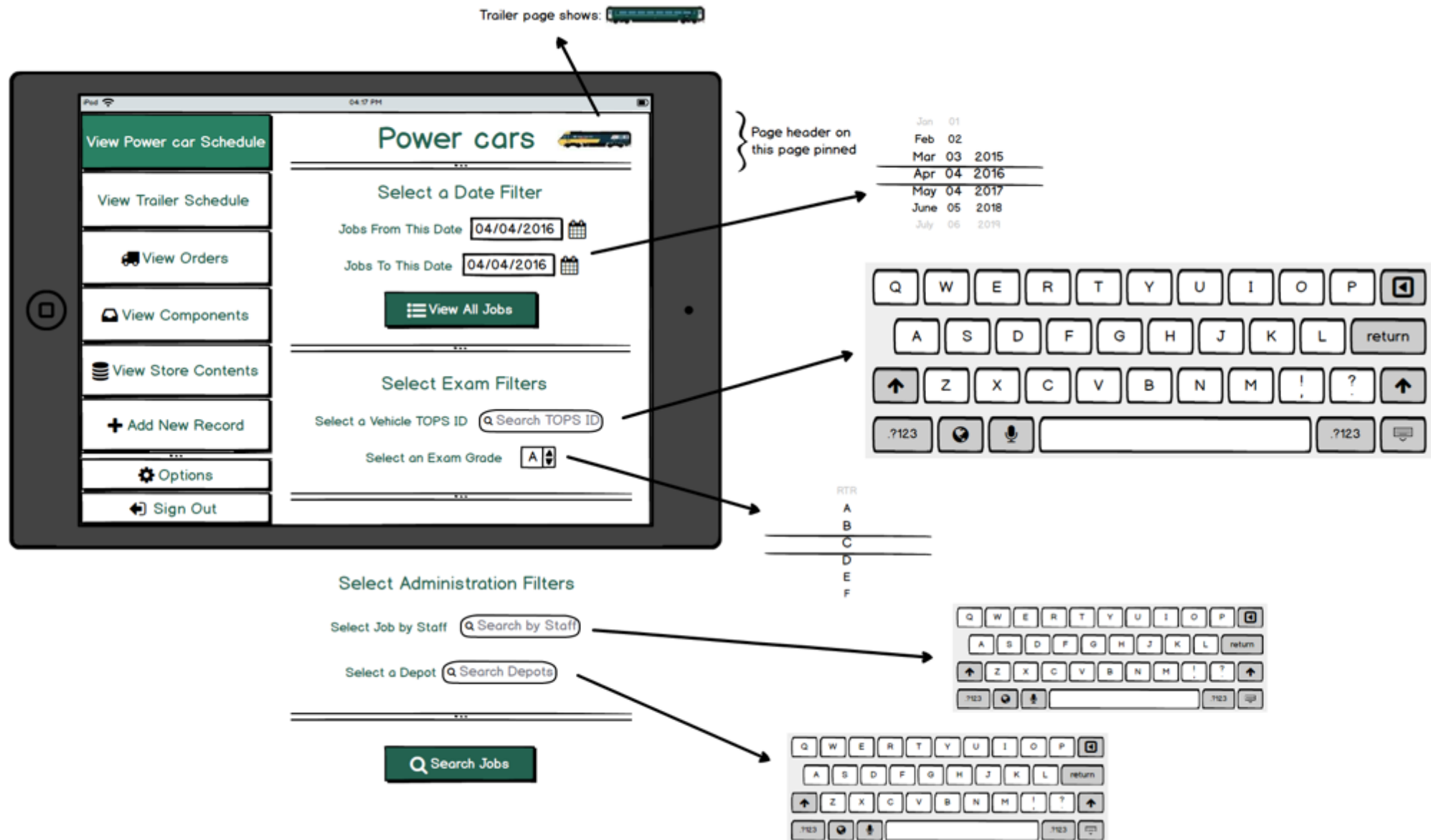


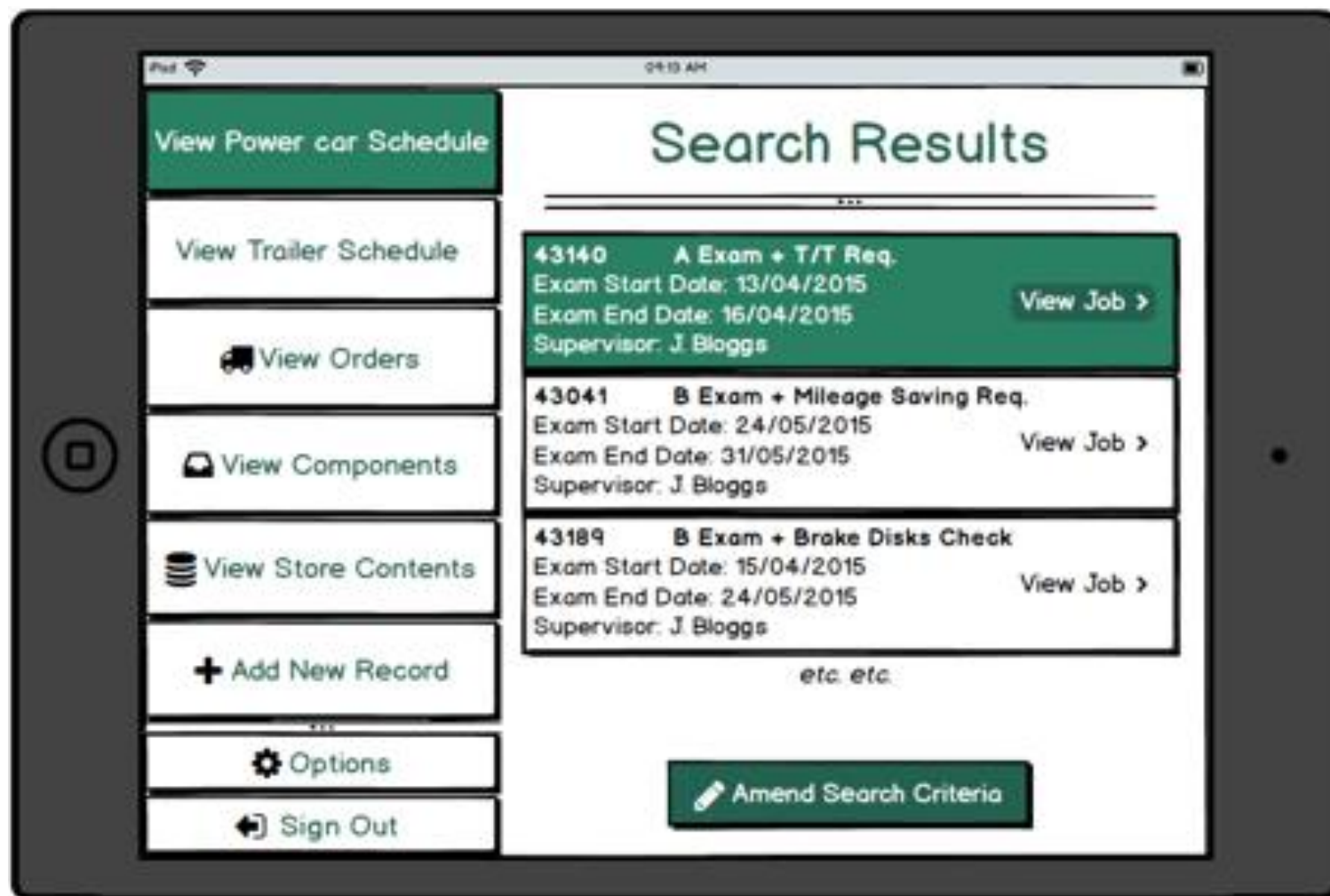




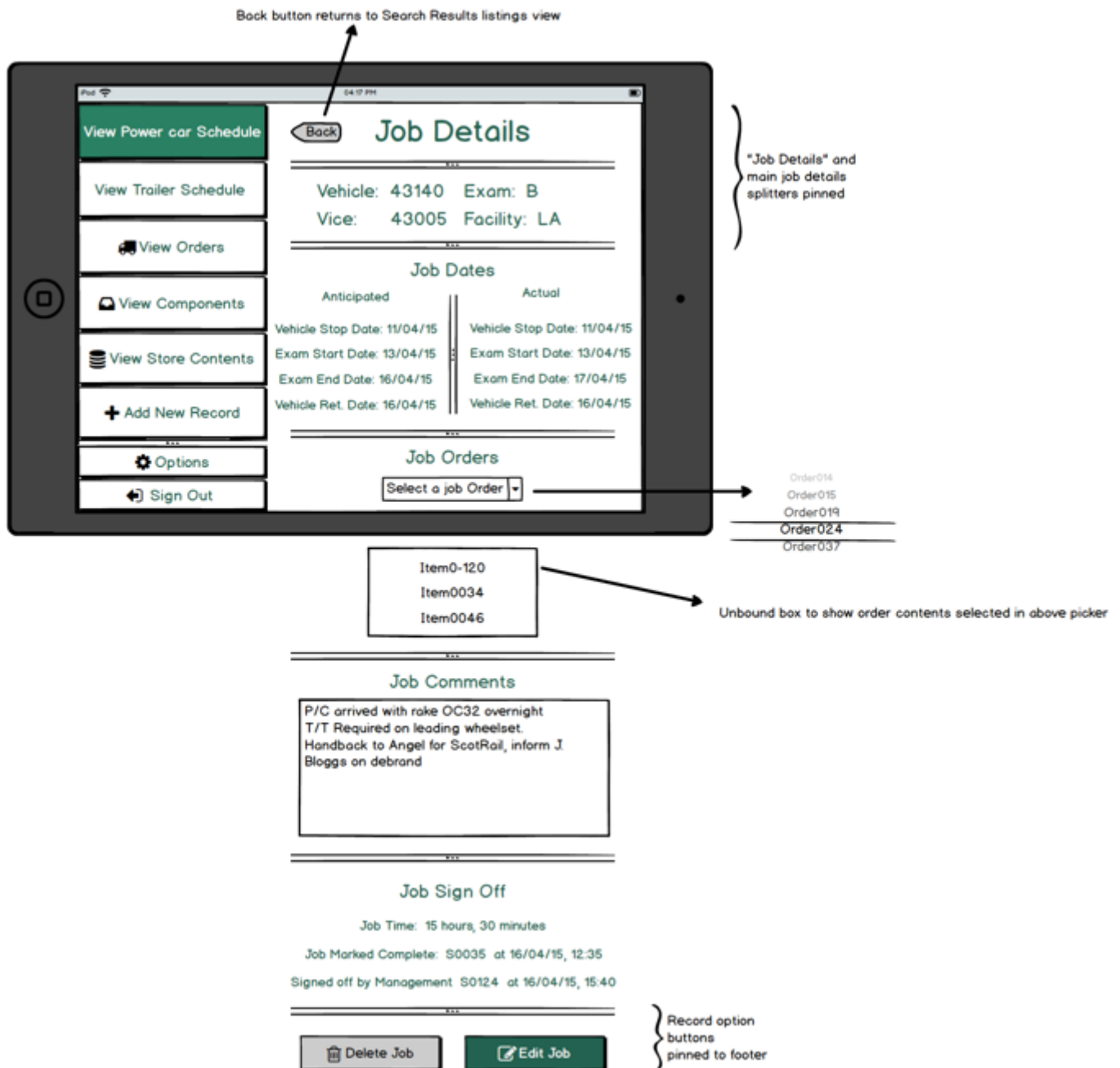


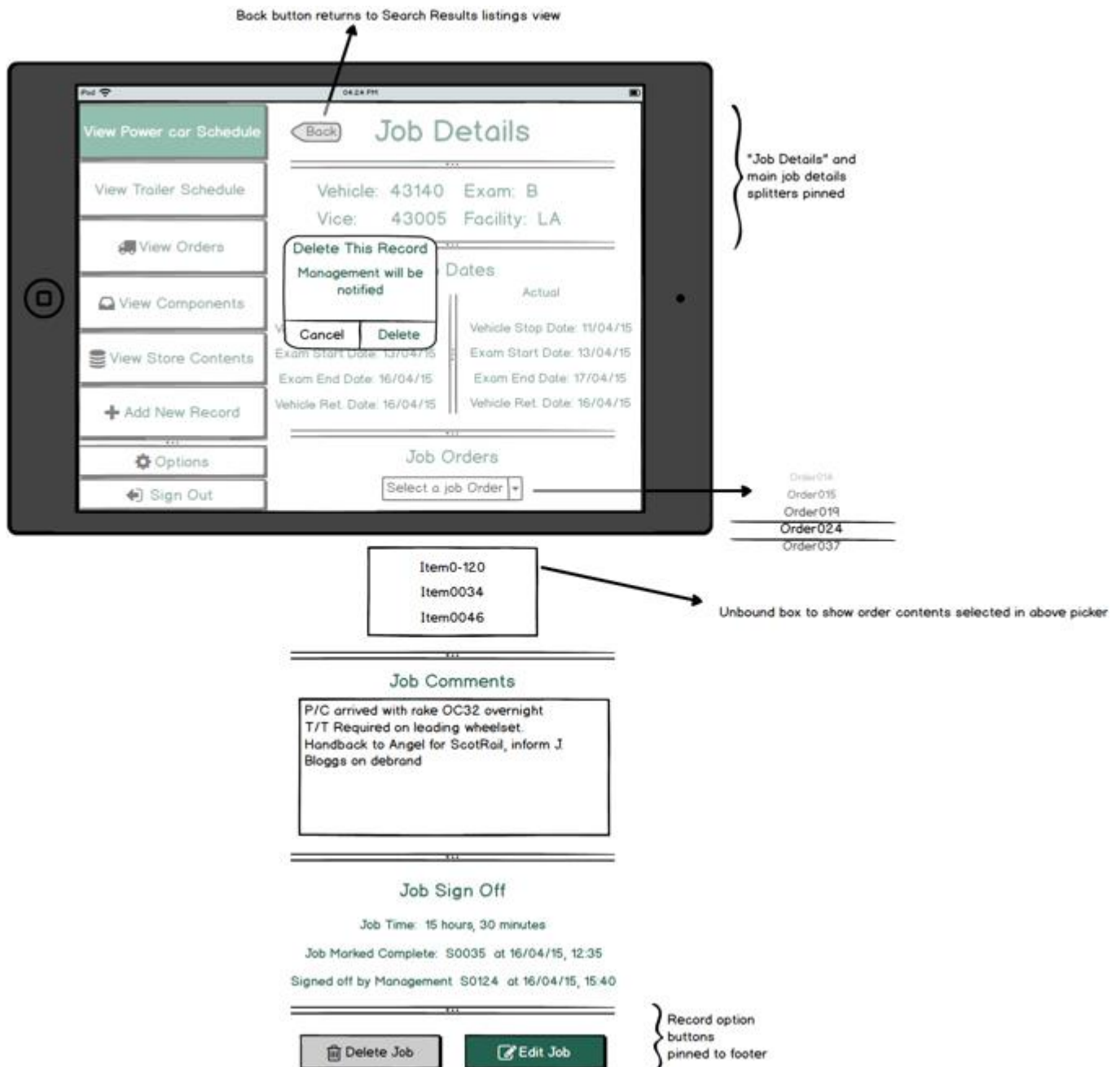


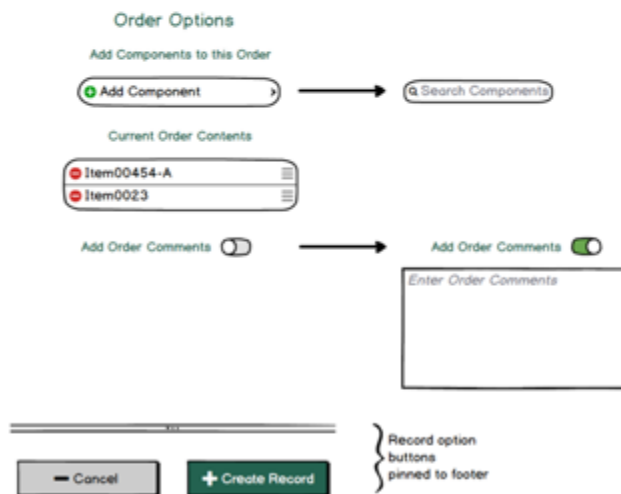
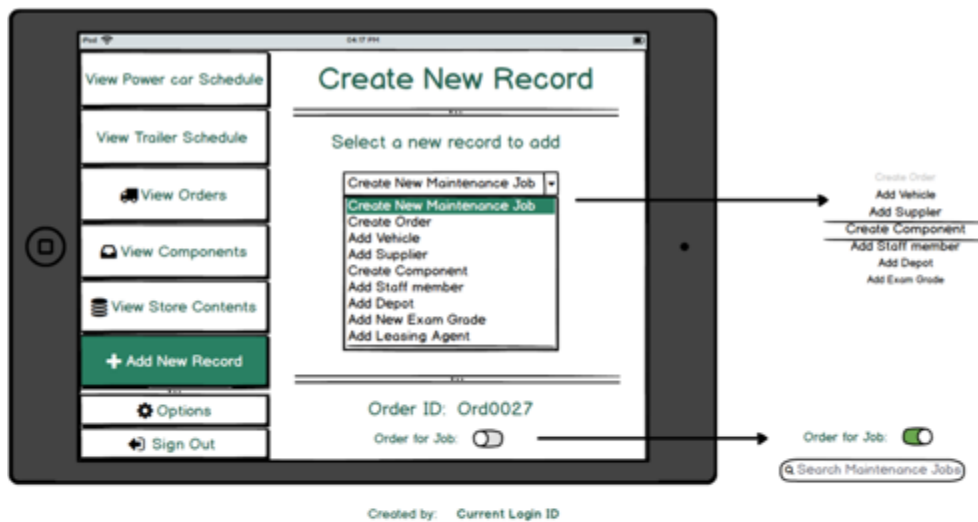




} "Search Results"
splitter pinned

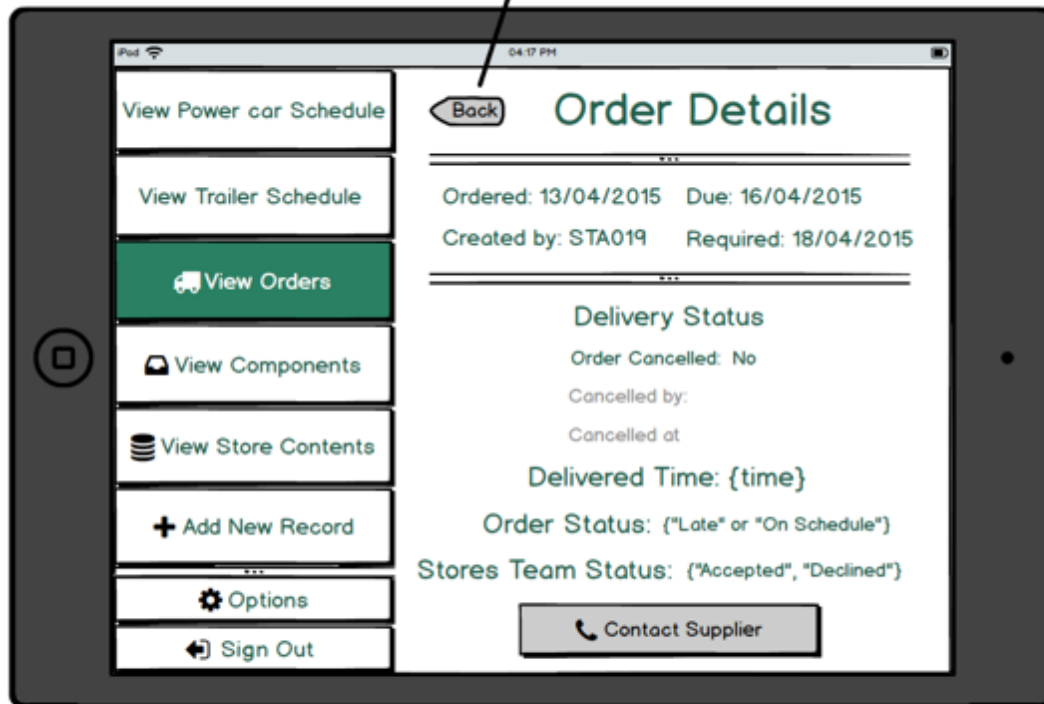






Shows an alert box with notification of complete action. The return to "Main Menu" page
Takes user back empty page with top combo-box showing tables where a new record can be added

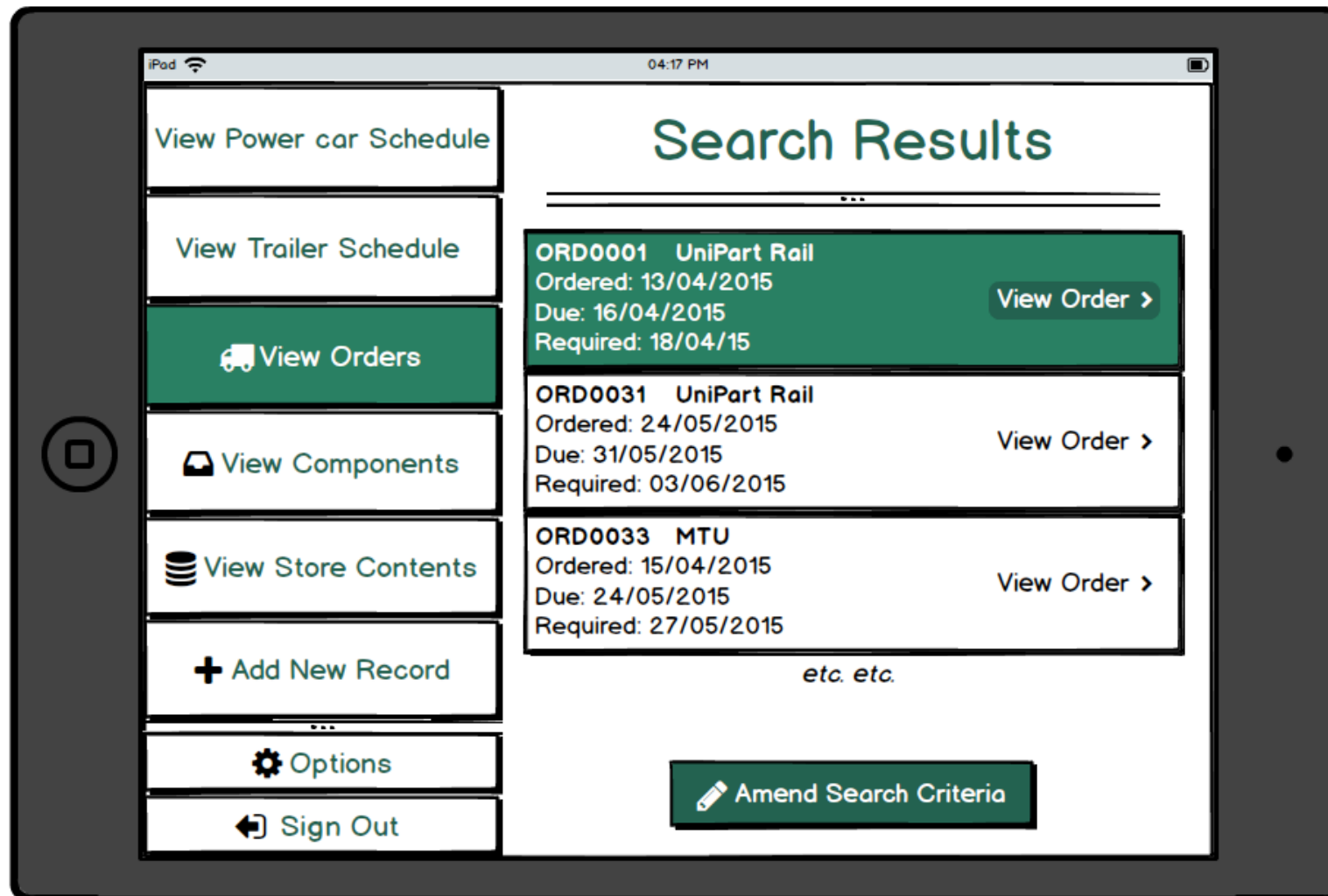
Back button returns to Search Results listings view



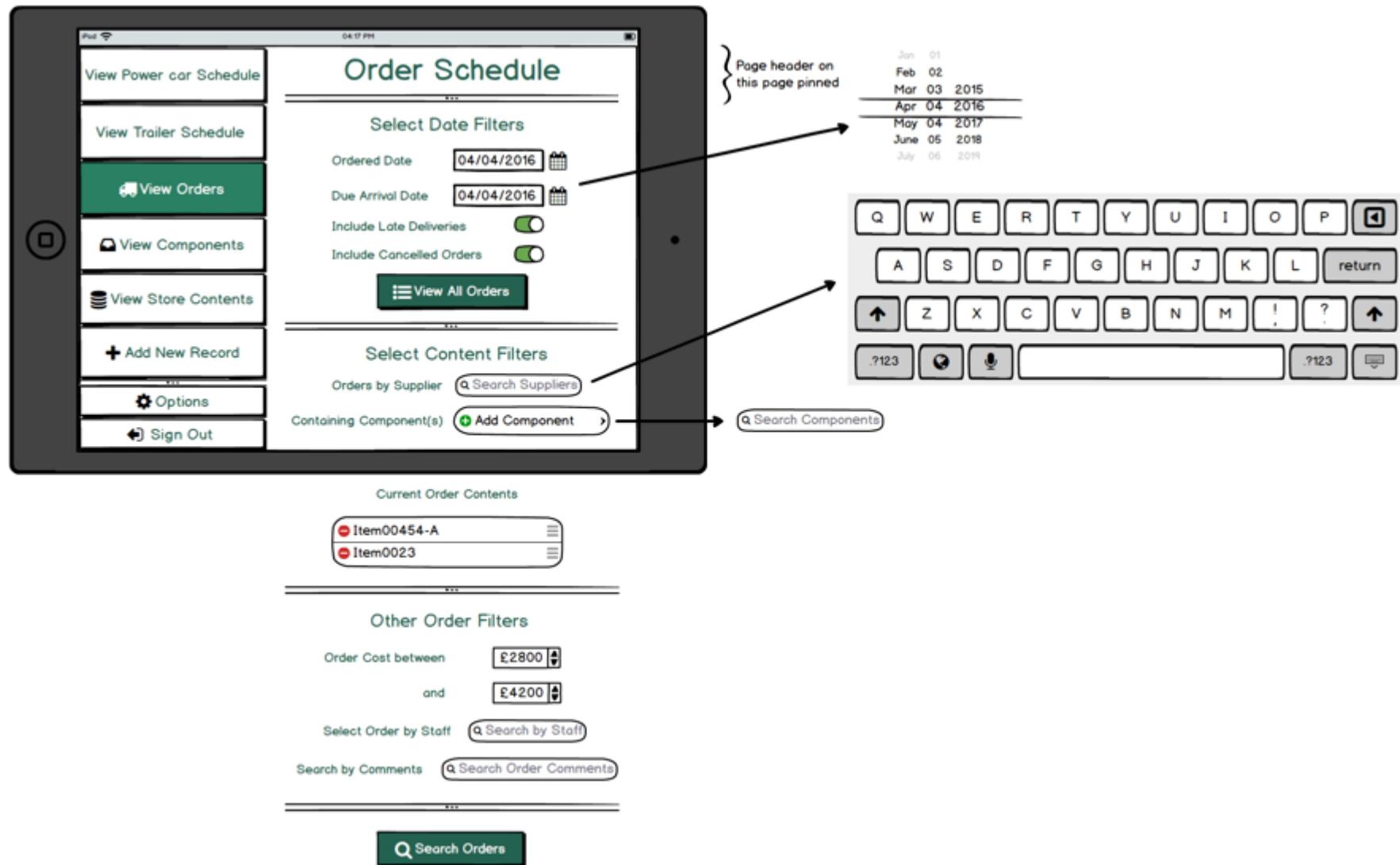
"Order Details" and
main job details
splitters pinned



Record option
buttons
pinned to footer



} "Search Results"
splitter pinned



10.5 Database SQL

10.5.1 Creating Tables

```
CREATE TABLE Staff(  
    Staff_ID CHAR(6) NOT NULL CONSTRAINT PK_Staff_ID PRIMARY KEY CHECK(Staff_ID  
        LIKE 'STA[0-9][0-9][0-9]'),  
    Staff_Surname VARCHAR(30) NOT NULL,  
    Staff_Forename VARCHAR(30) NOT NULL,  
    Staff_Contact_Email VARCHAR(35) NOT NULL,  
    Staff_Contact_Tele_No CHAR(12) NOT NULL  
        CHECK(Staff_Contact_Tele_No LIKE '0[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9]'), --standard UK format  
    Staff_Role VARCHAR(20) NOT NULL  
        CHECK(Staff_Role IN ('Lead Engineer', 'Engineer', 'Apprentice Engineer',  
            'Support', 'Admin', 'Team Leader', 'Manager', 'Site Manager')), --lists  
            job roles  
    Staff_Competency DEC(1,0) NOT NULL  
        CHECK(Staff_Competency IN ('1', '2', '3', '4', '5')), --1 is least  
            competent, 5 is most (e.g depot manager)  
    Staff_Site_Pin CHAR(27) NOT NULL  
        CHECK(Staff_Site_Pin LIKE '[A-Z][A-Z][0-9][0-9][0-9]'), --lists pin used  
            for high security depot entrance gates  
    Staff_Last_Pin_Change DATETIME NOT NULL, --date of last change  
    Comments VARCHAR(150),  
);  
  
--table working  
ALTER TABLE Staff  
ALTER COLUMN Comments VARCHAR(150);  
  
SELECT * FROM Staff;
```

```
CREATE TABLE Depot(  
    Depot_ID CHAR(2) NOT NULL CONSTRAINT PK_Depot_ID PRIMARY KEY,  
    Depot_Name VARCHAR(30) NOT NULL,  
    Depot_Address VARCHAR(75) NOT NULL,  
    Depot_Manager_ID CHAR(6) NOT NULL, FOREIGN KEY (Depot_Manager_ID)  
        REFERENCES Staff(Staff_ID),  
    Depot_No_Of_Roads DEC(2,0) NOT NULL,  
    Depot_Wheel_Lathe BIT,  
    Depot_Stores BIT,  
    Comments VARCHAR(150),  
);  
  
ALTER TABLE Depot  
ALTER COLUMN Depot_Address VARCHAR(75);  
  
ALTER TABLE Depot  
ALTER COLUMN Comments VARCHAR(150);
```

```

CREATE TABLE Vehicle(
  Vehicle_TOPS_ID VARCHAR(5) NOT NULL CONSTRAINT PK_Vehicle_TOPS_ID PRIMARY KEY,
  --Power cars and Trailers use 5-digit number, barrier coaches use 4-digit
  Vehicle_Depot_Allocation CHAR(2) NOT NULL, FOREIGN KEY (Vehicle_Depot_Allocation)
  REFERENCES Depot(Depot_ID), --its 'home' depot allocation
  Vehicle_Pool VARCHAR(4) NOT NULL
  CHECK(Vehicle_Pool IN ('EFOO', 'EFSH', 'EFPC', 'EFBV', 'EFTC', 'EFPZ')),
  --pool allocations used by firm in internal R2 database
  Vehicle_Status CHAR(1) NOT NULL CHECK(Vehicle_Status IN
  ('A', 'E', 'I', 'R', 'S')), --vehicle status code used by R2 stock listings
  database
  Vehicle_Config VARCHAR(5) CHECK(Vehicle_Config IN
  ('TF', 'TFD', 'TC', 'TRSMB', 'TRFB', 'TRSB', 'TS', 'TSD', 'TGS')), --
  applicable for trailers only - internal configuration of coach (e.g. 'TGS'
  = 'Trailer Guard Standard')
  Vehicle_Leasing_Agent CHAR(3) NOT NULL, FOREIGN KEY
  (Vehicle_Leasing_Agent) REFERENCES Leasing_Agent(Leasing_Agent_ID),
  Vehicle_Lease_Expires DATETIME NOT NULL,
  --date the vehicle *currently* is due to return to the parent leasing
  agent. Lease can be extended or terminated
  Vehicle_Restrictions VARCHAR(75), --any operational restrictions on main network
  Comments VARCHAR(150),
);

--pool allocations used by firm in internal R2 database

```

```

CREATE TABLE Leasing_Agent(
  Leasing_Agent_ID CHAR(3) NOT NULL CONSTRAINT PK_Leasing_Agent_ID PRIMARY KEY,
  Agent_Name VARCHAR(30) NOT NULL,
  Agent_Office_Address VARCHAR(75) NOT NULL,
  Agent_Contact_Name VARCHAR(30) NOT NULL,
  Agent_Contact_Email VARCHAR(75) NOT NULL,
  Agent_Contact_Tele_No CHAR(12) NOT NULL CHECK(Agent_Contact_Tele_No
  LIKE '0[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9]'),
  Comments VARCHAR(150),
);

ALTER TABLE Leasing_Agent
ALTER COLUMN Agent_Contact_Email VARCHAR(75);

```

```

CREATE TABLE Maintenance_Job(
  Job_ID CHAR(10) NOT NULL CONSTRAINT PK_Job_ID PRIMARY KEY CHECK(Job_ID
    LIKE 'MAINT[0-9][0-9][0-9][0-9][0-9]'),
  Job_Vehicle_No VARCHAR(5) NOT NULL, FOREIGN KEY (Job_Vehicle_No)
    REFERENCES Vehicle(Vehicle_TOPS_ID),
  Vice_Vehicle_No VARCHAR(5), FOREIGN KEY (Vice_Vehicle_No)
    REFERENCES Vehicle(Vehicle_TOPS_ID),
  Depot CHAR(2) NOT NULL, FOREIGN KEY (Depot) REFERENCES Depot(Depot_ID),
  Vehicle_Due_Stop_DateTime DATETIME,
    --nullable: vehicle stop could be corrective/unplanned
  Vehicle_Act_Stop_DateTime DATETIME,
    --nullable: vehicle might not actually stop for maintenance as scheduled
  Exam_Due_Start_DateTime DATETIME,
    --nullable: vehicle stop could be corrective/unplanned w/ no due start
  Exam_Act_Start_DateTime DATETIME,
    --nullable: vehicle might not actually stop for maintenance w/ no actual
    start date
  Exam_Due_End_DateTime DATETIME,
    --nullable: vehicle stop could be corrective w/ no anticipated end date
  Exam_Act_End_DateTime DATETIME,
    --nullable: vehicle might not actually stop for maintenance w/ no actual
    end date
  Vehicle_Due_Return_DateTime DATETIME, --might be no due return date
  Vehicle_Act_Return_DateTime DATETIME, --nullable: vehicle might not actually stop
    for maintenance
  Total_Job_Hours DEC(4,1) NOT NULL, --'0' if not undertaken. Only includes work
    times, not total vehicle stop time
  Staff_Signed_Off CHAR(6) NOT NULL, FOREIGN KEY (Staff_Signed_Off)
    REFERENCES Staff(Staff_ID),
  Signed_Off_DateTime DATETIME NOT NULL, --must have a signed off date,
    even if maintenance didn't occur
  Staff_Approved_Job CHAR(6) NOT NULL, FOREIGN KEY (Staff_Approved_Job)
    REFERENCES Staff(Staff_ID),
  Job_Comments VARCHAR(MAX), --comments field
  CONSTRAINT CHK_Vehicle_Swap
    CHECK(Vice_Vehicle_No != Job_Vehicle_No), /*check added to FK fields to ensure
    vehicle IDs are not the same, must be different vehicle going out (if a
    vehicle is returning after maintenance)*/
);

-- The 'actual' dates fields would be used for corrective jobs w/ 'due' columns left
null

```

```

CREATE TABLE Supplier(
    Supplier_ID CHAR(6) NOT NULL CONSTRAINT PK_SupplierID PRIMARY KEY
        CHECK(Supplier_ID LIKE 'SUP[0-9][0-9][0-9]'),
    Supplier_Name VARCHAR(30) NOT NULL,
    Supplier_Office_Address VARCHAR(75) NOT NULL,
    Supplier_Contact_Name VARCHAR(30) NOT NULL,
    Supplier_Contact_Tele_No CHAR(12) NOT NULL CHECK(Supplier_Contact_Tele_No
        LIKE '0[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9]'),
    Supplier_Contact_Email VARCHAR(75) NOT NULL,
    Comments VARCHAR(50), --supplier comments
);

ALTER TABLE Supplier
ALTER COLUMN Supplier_Office_Address VARCHAR(75);

```

```

CREATE TABLE Grade(
    Grade_ID VARCHAR(2) NOT NULL CONSTRAINT PK_Grade_ID PRIMARY KEY,
    Advertised_Grade_Hours DEC(4,1) NOT NULL,
    Store_Location VARCHAR(2) NOT NULL,
        --lists location where item could be found in stores. Exam Packs still
        need location listed
    Comments VARCHAR(150),
);

ALTER TABLE Grade
ALTER COLUMN Advertised_Grade_Hours DEC(4,1);

```

```

CREATE TABLE Component(
    Component_ID VARCHAR(30) NOT NULL CONSTRAINT Pk_Component_ID PRIMARY KEY,
        --no check, large PK field as ID used is the suppliers for the piece.
        Assume no re-numbering takes place
    Component_Name VARCHAR(35) NOT NULL,
    Component_Agent CHAR(3) NOT NULL, FOREIGN KEY (Component_Agent) REFERENCES
        Leasing_Agent(Leasing_Agent_ID), --agent recommends package
    Component_Grade VARCHAR(2) NOT NULL, FOREIGN KEY (Component_Grade)
        REFERENCES Grade(Grade_ID), --package is used in a maintenance job
    Component_Cost DEC(6,2) NOT NULL,
    Quantity_In_Stock DEC(3,0) NOT NULL,
    Comments VARCHAR(150), --Component or package comments
);

```

```

CREATE TABLE Job_Order(
  Order_ID CHAR(7) NOT NULL CONSTRAINT PK_Order_ID PRIMARY KEY
    CHECK(Order_ID LIKE 'Ord[0-9][0-9][0-9][0-9]'),
  Job_ID CHAR(10) NOT NULL, FOREIGN KEY (Job_ID)
    REFERENCES Maintenance_Job(Job_ID),
  Order_Staff_ID CHAR(6) NOT NULL, FOREIGN KEY (Order_Staff_ID)
    REFERENCES Staff(Staff_ID), --who created order
  Order_Supplier_ID CHAR(6) NOT NULL, FOREIGN KEY (Order_Supplier_ID)
    REFERENCES Supplier(Supplier_ID), --supplier for order
  Order_Created_DateTime DATETIME NOT NULL,
  Order_Cancelled_Staff_ID CHAR(6), FOREIGN KEY
    (Order_Cancelled_Staff_ID) REFERENCES Staff(Staff_ID), --if order
    cancelled, by whom
  Order_Cancelled_DateTime DATETIME, --time of cancellation
  Order_Required_DateTime DATETIME NOT NULL, --no null, always be a date for a job
    to commence, might not now need order as listed in previous fields
  Order_Delivered_DateTime DATETIME NOT NULL, --when was ordered delivered to depot
  Order_Cost DECIMAL(6,2), --order cost
  Comments VARCHAR(50),
);

```

```

CREATE TABLE Order_Component(
  Order_ID CHAR(7) NOT NULL, FOREIGN KEY (Order_ID) REFERENCES
    Job_Order(Order_ID), --job order ID field
  Component_ID VARCHAR(30) NOT NULL, FOREIGN KEY (Component_ID) REFERENCES
    Component(Component_ID), --placed against component ID(s)
  Comments VARCHAR(50), --comments on contents of component order
);

```

```

CREATE TABLE WholeSystemLog(
    LogID [INT] IDENTITY(1,1) NOT NULL CONSTRAINT Pk_Log_ID PRIMARY KEY,
        --unique ID for each change in a database contents
    DatabaseName VARCHAR(256) NOT NULL, --in which db did this change occur?
    EventType VARCHAR(50) NOT NULL, --"ADD / ALTER / DROP" then type
        [_procedure/table/function/view]
    ObjectName VARCHAR(256) NOT NULL, --procedure name, table name, function
        name or view name
    ObjectType VARCHAR(25) NOT NULL, --its either a procedure, table, function,
        view
    TSQLCommand VARCHAR(MAX) NOT NULL, --copies in the code that was ran, (MAX)
        size of up to 2GB rather than 256
    EventDateTime DATETIME NOT NULL CONSTRAINT Df_Events_Log_GetDateTime
        DEFAULT (GetDate()), --type of eventTime is DateTime, set 'GetDate()'
        as field format
    LoginName VARCHAR(256) NOT NULL --database user running tsqlcommand
);

--Used to alter a table, then view contents of log file.
SELECT * FROM WholeSystemLog;

ALTER TABLE Staff DROP COLUMN TestColumn;

UPDATE Staff
SET Comments = null
WHERE Staff_ID = 'STA001';

SELECT * FROM Staff;

DROP TABLE WholeSystemLog;

SELECT * FROM Supplier;

```

10.5.2 Creating Records

Only a selection of inserted table records are shown here for appendix clarity. Please refer to attached file submission for complete INSERT commands.

```

INSERT INTO Staff(Staff_Surname, Staff_Forename, Staff_Contact_Email,
    Staff_Contact_Tele_No, Staff_Role, Staff_Competency, Staff_Site_Pin,
    Staff_Last_Pin_Change, Comments)

VALUES ('Williams', 'Mark', 'mark.williams@gwr.com', '07787-156442', 'Engineer', '3',
    'TF090', '12-Dec-2018 20:30', null);

INSERT INTO Staff(Staff_Surname, Staff_Forename, Staff_Contact_Email,
    Staff_Contact_Tele_No, Staff_Role, Staff_Competency, Staff_Site_Pin,
    Staff_Last_Pin_Change, Comments)

VALUES ('Edwards', 'Ben', 'ben.edwards@gwr.com', '07462-008786', 'Lead Engineer', '4',
    'FD228', '04-Nov-2018 14:42', null);

```

```
--P/Cs input by TOPS ID
INSERT INTO Vehicle (Vehicle_TOPS_ID, Vehicle_Depot_Allocation, Vehicle_Pool,
Vehicle_Status, Vehicle_Config, Vehicle_Leasing_Agent, Vehicle_Lease_Expires,
Vehicle_Restrictions, Comments)

VALUES ('43002', 'LA', 'EFPC', 'A', null, 'PTR', '04-Jul-2019 00:00', null, 'speak to NRM
reg. preservation');

--OC37 (as of 20/02) showing only TGS 42575
INSERT INTO Vehicle (Vehicle_TOPS_ID, Vehicle_Depot_Allocation, Vehicle_Pool,
Vehicle_Status, Vehicle_Config, Vehicle_Leasing_Agent, Vehicle_Lease_Expires,
Vehicle_Restrictions, Comments)

VALUES ('42575', 'LA', 'EFTC', 'A', 'TGS', 'ANG', '19-Oct-2019 00:00', null, 'only
vehicle allox rake OC37');
```

```
--Engines
INSERT INTO Component (Component_ID, Component_Name, Component_Agent, Component_Grade,
Component_Cost, Quantity_In_Stock, Comments) VALUES ('MTU_1557a', 'MTU 8V 4000
R43', 'ANG', 'F', '75000.99', '0', null); --MTU

--Traction and Gearing
INSERT INTO Component (Component_ID, Component_Name, Component_Agent, Component_Grade,
Component_Cost, Quantity_In_Stock, Comments) VALUES ('114-2F', 'Traction
converter', 'ANG', 'D', '1100.99', '0', 'Reauthentication due 12/05');
```

```
/* A Exam */
INSERT INTO Maintenance_Job (Job_Vehicle_No, Vice_Vehicle_No, Depot,
Vehicle_Due_Stop_DateTime, Vehicle_Act_Stop_DateTime, Exam_Due_Start_DateTime,
Exam_Act_Start_DateTime, Exam_Due_End_DateTime, Exam_Act_End_DateTime,
Vehicle_Due_Return_DateTime, Vehicle_Act_Return_DateTime, Total_Job_Hours,
Staff_Signed_Off, Signed_Off_DateTime, Staff_Approved_Job, Job_Comments)

VALUES ('43010', '43196', 'LA', '26-Oct-2018 02:00', '25-Mar-2018 11:00', '26-Mar-2018
15:00', '26-Mar-2018 09:00', '26-Mar-2018 00:00', '26-Mar-2018 11:00', '27-Mar-
2018 02:00', '27-March-2018 02:00', '7.5', 9, '26-Mar-2018 00:00', 13, null);

/* Corrective Job */
INSERT INTO Maintenance_Job (Job_Vehicle_No, Vice_Vehicle_No, Depot,
Vehicle_Due_Stop_DateTime, Vehicle_Act_Stop_DateTime, Exam_Due_Start_DateTime,
Exam_Act_Start_DateTime, Exam_Due_End_DateTime, Exam_Act_End_DateTime,
Vehicle_Due_Return_DateTime, Vehicle_Act_Return_DateTime, Total_Job_Hours,
Staff_Signed_Off, Signed_Off_DateTime, Staff_Approved_Job, Job_Comments)

VALUES ('43140', '43087', 'LA', null, '26-Mar-2018 22:00', null, '27-Mar-2018 02:00',
null, '27-Mar-2018 12:00', null, '27-Mar-2018 17:30', '5.5', 1, '27-Mar-2018
19:25', 12, null);
```

```

--- Maintenance Record 1 ---
INSERT INTO Job_Order (Job_ID, Order_Staff_ID, Order_Supplier_ID, Order_Created_DateTime,
    Order_Cancelled_Staff_ID, Order_Cancelled_DateTime, Order_Required_DateTime,
    Order_Delivered_DateTime, Order_Cost, Comments)

VALUES (1, 4, 2, '20-Mar-2018 12:20', null, null, '25-Mar-2018 22:00', '24-Mar-2018
    17:30', '575.59', null);

INSERT INTO Job_Order (Job_ID, Order_Staff_ID, Order_Supplier_ID, Order_Created_DateTime,
    Order_Cancelled_Staff_ID, Order_Cancelled_DateTime, Order_Required_DateTime,
    Order_Delivered_DateTime, Order_Cost, Comments)

VALUES (1, 9, 5, '18-Mar-2018 22:20', null, null, '25-Mar-2018 22:00', '22-Mar-2018
    08:15', '1238.50', null);

INSERT INTO Job_Order (Job_ID, Order_Staff_ID, Order_Supplier_ID, Order_Created_DateTime,
    Order_Cancelled_Staff_ID, Order_Cancelled_DateTime, Order_Required_DateTime,
    Order_Delivered_DateTime, Order_Cost, Comments)

VALUES (1, 9, 4, '22-Mar-2018 19:50', 9, '23-Mar-2018 00:30', '25-Mar-2018 22:00', '25-
    Mar-2018 10:15', '81210.99', 'cancelled - traction motor order + change postponed
    til 30-11-2018'); --order cancelled.

```

```

--- Job Order 1 ---
INSERT INTO Order_Component(Order_ID, Component_ID, Comments) VALUES (1, '1602/181101',
    null);

INSERT INTO Order_Component(Order_ID, Component_ID, Comments) VALUES (1, '0017/007200',
    null);

INSERT INTO Order_Component(Order_ID, Component_ID, Comments) VALUES (1, '151956-df',
    null);

```

10.5.3 Queries, Triggers, Test and Configuration SQL

This Appendix section includes all of the additional database queries, triggers and transaction statements executed on the database.

```

SELECT * FROM dbo.Vehicle; -- view table

SELECT * FROM Leasing_Agent;

SELECT * FROM Job_Order;

SELECT * FROM sys.event_log
ORDER BY start_time DESC;

-- Server=tcp:majorprojectdatabase.database.windows.net,1433;Initial
  Catalog=Laira_Vehicle_Maintenance;Persist Security Info=False;User
  ID=Henry;Password=Administrator_Pa55;MultipleActiveResultSets=False;Encrypt=True;T
  rustServerCertificate=False;Connection Timeout=30;

CREATE USER henrycobbold FROM LOGIN henrycobbold;
SELECT * FROM sysusers;

SELECT * FROM sys.database_principals;

CREATE LOGIN Henry WITH PASSWORD = 'Administrator_Pa55';
CREATE USER Henry FROM LOGIN Henry;

SELECT * FROM WholeSystemLog
WHERE EventType = 'ALTER_TABLE';

SELECT * FROM Grade;

SELECT * FROM Supplier;

ALTER ROLE db_datawriter ADD MEMBER Henry;
ALTER ROLE db_datareader ADD MEMBER Henry;
ALTER ROLE db_ddladmin ADD MEMBER Henry;

SELECT * FROM Maintenance_Job;
SELECT * FROM Job_Order;

```

```

--ALTER ROLE db_datawriter ADD MEMBER henrycobbold;

--Run attempt failed, job has been rescheduled. Last error was: Incorrect syntax near the
keyword 'FROM'.

--jobs for a vehicle
SELECT Job_Vehicle_No, Vice_Vehicle_No, Exam_Act_Start_DateTime, Exam_Act_End_DateTime,
       Total_Job_Hours, Staff_Signed_Off, Signed_Off_DateTime
FROM Maintenance_Job
WHERE Job_Vehicle_No = '43140';

--jobs completed between certain dates ordered by their length
SELECT * FROM Maintenance_Job WHERE Exam_Act_End_DateTime BETWEEN '01-Apr-2018' AND '10-
Apr-2018' ORDER BY Total_Job_Hours ASC;

--exams that started on site later than they were originally scheduled to
SELECT Job_ID, Depot, Exam_Due_Start_DateTime, Exam_Act_Start_DateTime
FROM Maintenance_Job
WHERE Exam_Act_Start_DateTime >= Exam_Due_Start_DateTime ORDER BY Job_ID ASC;

--jobs signed off more than 3 days after they were said to have been completed (i.e. a
late sign off by staff that causes unnecessary delay)
SELECT Job_ID, Job_Vehicle_No, Exam_Act_End_DateTime, Signed_Off_DateTime,
       DATEDIFF(hour, Exam_Act_End_DateTime, Signed_Off_DateTime) AS Hours_After_Signature
FROM Maintenance_Job
WHERE DATEDIFF(hour, Exam_Act_End_DateTime, Signed_Off_DateTime) > 5 ORDER BY
Job_ID;

--no. of items contained in each order
SELECT Order_ID AS [Order_Number], COUNT(Component_ID) AS No_Of_Items_In_Order
FROM Order_Component
GROUP BY Order_ID;

--select orders that were delivered late sorted by date
SELECT Order_ID, Order_Created_DateTime, Order_Required_DateTime,
       Order_Delivered_DateTime
FROM Job_Order
WHERE Order_Required_DateTime <= Order_Delivered_DateTime
ORDER BY Order_Created_DateTime DESC;

UPDATE Maintenance_Job
SET Job_Comments = null
WHERE Job_ID = 13;

ALTER TABLE Job_Order
ADD Order_Accepted_DateTime DATETIME; --nullable to accept rejected orders

```

```

USE [Laira_Vehicle_Maintenance]
GO

CREATE TRIGGER Trig_WholeSystemAuditLog
ON DATABASE
FOR CREATE_PROCEDURE,
    ALTER_PROCEDURE,
    DROP_PROCEDURE,
    CREATE_TABLE,
    ALTER_TABLE,
    DROP_TABLE,
    CREATE_FUNCTION,
    ALTER_FUNCTION,
    DROP_FUNCTION,
    CREATE_VIEW,
    ALTER_VIEW,
    DROP_VIEW
AS

SET NOCOUNT ON

DECLARE @data XML
SET @data = EVENTDATA()

INSERT INTO WholeSystemLog(DatabaseName, EventType,
    ObjectName, ObjectType, TSQLCommand, LoginName)
VALUES(
@data.value('(/EVENT_INSTANCE/DatabaseName)[1]', 'varchar(256)'),
@data.value('(/EVENT_INSTANCE/EventType)[1]', 'varchar(50)'),
@data.value('(/EVENT_INSTANCE/ObjectName)[1]', 'varchar(256)'),
@data.value('(/EVENT_INSTANCE/ObjectType)[1]', 'varchar(25)'),
@data.value('(/EVENT_INSTANCE/TSQLCommand)[1]', 'varchar(max)'),
@data.value('(/EVENT_INSTANCE/LoginName)[1]', 'varchar(256)')
);

GO
ENABLE TRIGGER Trig_WholeSystemAuditLog ON DATABASE;

```

10.6 Wireframe Testing: Cognitive Walkthrough Evaluation Outcomes

The completed Cognitive Walkthrough tables filled out during Heuristic Evaluation.

10.6.1 Walkthrough A: Marking a Maintenance Job Complete

1. Sign in using “Staff ID” and “Password”

Table 21: Task A: Step 1 Test Outcomes

<i>Current Screen View: “Login”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, users are aware that they need to use their identification and authentication to sign themselves into the app.
Can the user see a control to do that?	Yes	The boxes for “Staff ID” (identification) and “Password” (authentication) are clearly obvious. There is also a large “Sign In” button.
Is it obvious how the control works?	Yes	There is one button and two text fields that when touched bring up a keyboard for text entry.
Does the user understand the feedback they have been given?	Yes	Having entered the correct details, the following screen shows a two-pane window with menu items on the left if the details are correct. If the attempt was unsuccessful, a clear pop-up shows the failed attempt.

2. Select the Power car vehicle pane

Table 22: Task A: Step 2 Test Outcomes

<i>Current Screen View: “Main Menu”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, the user knows that they want to view the power car schedule because an engineer knows that they want to edit an existing job entry to mark it complete.
Can the user see a control to do that?	Yes	A tab showing “View Power car Schedule” looks likely. Options to view orders, components, stores or add a new record don’t look as likely as this first option.
Is it obvious how the control works?	Yes	The box has nothing except text and is clearly outlined, and the right-hand pane reads “Select a Menu Item”. It

		should be very clear that this control is a button.
Does the user understand the feedback they have been given?	Yes	Yes, it is obvious what selecting this control has done. The tab has now turned green and the right pane is showing the title “Power cars” with a range of fields below.

3. Select the appropriate date filters

Table 23: Task A: Step 3 Test Outcomes

<i>Current Screen View: “Power cars”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, there is a sub-heading saying “Select a Date Filter” that instructs the user to enter some date parameters. The user understands that this will help to filter the job results returned.
Can the user see a control to do that?	Yes	Yes, there are two main controls in this “table” (section of page) to define a date range. The calendar symbol helps improve clarity.
Is it obvious how the control works?	Yes	Yes, there is a box with a calendar symbol alongside as well as the current date inside the box. This clearly implies that a day can be selected by clicking here.
Does the user understand the feedback they have been given?	Yes	A date picker spinner appears when clicking on the control. The month, day and year are obviously selectable by scrolling through the wheel. The “Done” button alongside <u>that is part of this control</u> clearly allows the user to exit the spinner having highlighted the desired date.

4. Select appropriate exam filters

Table 24: Task A: Step 4 Test Retest Outcomes

<i>Current Screen View: “Power cars”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, the user clearly understands if they want to apply any filters or not – all filter controls on this page are optional.
Can the user see a control to do that?	Yes	Yes, the user can see an option to filter results by vehicle and by exam grade in another page “table”.

Is it obvious how the control works?	Yes	Yes. Despite much consideration this control could be more quickly understood by system users if it were change into a text box. This decision however will require further thought and evaluation.
Does the user understand the feedback they have been given?	Yes	Yes, the vehicle search control brings up the device keyboard for users to enter the vehicle TOPS ID, typing values brings up search results below the field. Clear and obvious what this action has done. Exam grade field brings up the spinner view at the bottom of the screen so is suitable.

5. Select the appropriate administration filters

Table 25: Task A: Step 5 Test **Retest** Outcomes

<i>Current Screen View: "Power cars"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	They can see that they have the option to select jobs in the schedule by the staff member assigned to them or the site at which they have been carried out.
Can the user see a control to do that?	Yes	Yes, there are clearly several controls that users can alter to set the location and staff with the control label
Is it obvious how the control works?	Yes	Yes, they are both text fields and so are familiar and easily recognisable. The placeholder text also makes this clear.
Does the user understand the feedback they have been given?	Yes	Now both "Select Job by Staff" and "Select a Depot" show as text boxes and are more predictable for users.

6. Search for records matching the applied filters

Table 26: Task A: Step 6 Test Outcomes

<i>Current Screen View: "Power cars"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, they want to search having selected the desired filters.

Can the user see a control to do that?	Yes	Yes, a large button at the bottom of filter options reads <i>"Search Jobs"</i> . This is clearly the control to perform this action.
Is it obvious how the control works?	Yes	Yes, clearly outlined and widely familiar.
Does the user understand the feedback they have been given?	Yes	Yes, <i>"Search Results"</i> are now presented to the user. This has clearly performed the search action.

7. Find the desired job from the results list returned

Table 27: Test A: Step 7 Test Outcomes

<i>Current Screen View: "Search Results"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, a user wishes to select the correct job from the list, allowing them to update its information and mark it as complete.
Can the user see a control to do that?	Yes	Yes, if the filters applied were accurate then the correct job (and others that meet the criteria) will be displayed. Each job shows <i>"View Job"</i> , implying that it can be clicked on to view it in more detail.
Is it obvious how the control works?	Yes	The arrow icon on the result implies that it can be clicked on to open up a new view of the record.
Does the user understand the feedback they have been given?	Yes	User is presented with <i>"Job Details"</i> so the result of selecting the record is clearly evident.

8. Edit the job to complete the empty fields to mark as complete

Table 28: Test A: Step 8 Test Retest Outcomes

<i>Current Screen View: "Job Details"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, the user understands they want to edit the information to update the jobs status.

Can the user see a control to do that?	Yes	Not at the top of the screen but scrolling to the bottom shows the option to “Edit Job” or “Delete Job”. The user wishes to edit so a control is plainly visible here that looks like it will allow this action.
Is it obvious how the control works?	Yes	Yes, it’s a button that is widely understood.
Does the user understand the feedback they have been given?	Yes	This control has been replaced by a simple static list showing the Order ID and the Component IDs underneath. This new control will be able to easily display any number of orders for each job.

9. Save and close the job

Table 29: Task A: Step 9 Test Outcomes

Current Screen View: “Job Details”		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	Yes	Yes, they are aware that they must save the changes made to the job record.
Can the user see a control to do that?	Yes	A Large “Save Record” button has appeared at the bottom of the page alongside a “Cancel” button. They wish to save so this is a clear option.
Is it obvious how the control works?	Yes	Yes, it’s a large green button – perfectly obvious.
Does the user understand the feedback they have been given?	Yes	They have returned to the job information and non-editable fields. The information contained within the record has updated. They have saved the changes made to the record.

10. Sign out of the program

Table 30: Task A: Step 10 Test Retest Outcomes

Current Screen View: “Job Details”		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	Yes	This is now clear ... Each time the app launches the user is reminded that they must actively sign out before they close the application. This is a security enforcement rule.

Can the user see a control to do that?	Yes	There is a <i>"Sign Out"</i> button visible at the bottom right of the screen. This looks like a way to sign out of an account. The accompanying icon helps make its purpose clear.
Is it obvious how the control works?	Yes	It's a button, so is a widely familiar control.
Does the user understand the feedback they have been given?	Yes	They have returned to the 'Sign In' screen and can see an option to sign into the app. They are aware that this means they have successfully signed out.

10.6.2 Walkthrough B: Edit an Existing Component Order

1. Sign in with “Staff ID” and “Password”

Table 31: Task B: Step 1 Test Outcomes

Current Screen View: “Login”		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	Yes	Yes, users are aware that they need to use their identification and authentication to sign themselves into the app.
Can the user see a control to do that?	Yes	The boxes for “Staff ID” (identification) and “Password” (authentication) are clearly obvious. There is also a large “Sign In” button.
Is it obvious how the control works?	Yes	There is one button and two text fields that when touched bring up a keyboard for text entry.
Does the user understand the feedback they have been given?	Yes	Having entered the correct details, the following screen shows a two-pane window with menu items on the left if the details are correct. If the attempt was unsuccessful, a clear pop-up shows the failed attempt.

2. Select the “View Orders” menu pane

Table 32: Task B: Step 2 Test Outcomes

Current Screen View: “Main Menu”		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	Yes	Yes, they wish to edit the contents of an order that has already been placed.
Can the user see a control to do that?	Yes	There is a “View Orders” button in the left-hand pane that looks like it will show the orders that have been placed.
Is it obvious how the control works?	Yes	Yes, it is a button and is clearly placed alongside other buttons. There is no confusing this control.
Does the user understand the feedback they have been given?	Yes	It is clear that pressing this button has taken the user to a completely different page. The screen now shows several fields to refine the order records returned to the user. Clearly this button has produced a search feature and the

		user understands that this button has shown this screen.
--	--	--

3. Select (if suitable) the appropriate date filters to find the given order

Table 33: Task B: Step 3 Test Retest Outcomes

<i>Current Screen View: "Order Schedule"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, there is a sub-heading saying " <i>Select a Date Filter</i> " that instructs the user to enter some date parameters. The user understands that this will help to filter the order results returned.
Can the user see a control to do that?	Yes	Yes, there are two main controls in this "table" (section of page) to define a date range. The calendar symbol helps improve clarity.
Is it obvious how the control works?	Yes	The selection of checkboxes here is more suitable for a web environment and has been replaced by iOS style switches that offer a simple "Yes" or "No" value.
Does the user understand the feedback they have been given?	Yes	A date picker spinner appears when clicking on the control. The month, day and year are obviously selectable by scrolling through the wheel. The "Done" button alongside <u>that is part of this control</u> clearly allows the user to exit the spinner having highlighted the desired date. A checkbox (or alternative switch) clearly shows a different state and the result of changing this binary setting.

4. Select (if suitable) the appropriate "Content" filters to find the given order

Table 34: Task B: Step 4 Test Outcomes

<i>Current Screen View: "Order Schedule"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes. The user is aware that they have the option to find the order that they wish to update by searching all records that contain certain components.
Can the user see a control to do that?	Yes	There are several controls that allow filtering by order supplier and orders " <i>Containing Component(s)</i> ". A search box to select a supplier and a search list option to select a component are clearly visible.

Is it obvious how the control works?	Yes	The search box for supplier is widely accepted and understood. No concerns with this choice. The “Add Component” button, taking the user to the “Search Components” option is easily understandable and will allow components to added to the search option.
Does the user understand the feedback they have been given?	Yes	Yes, a supplier is visible in the field or a component can be seen in the contents search list. A search can now be performed that looks explicitly for the given combination of supplier and/or component(s)

5. Select (if suitable) the appropriate other filters to find the given order

Table 35: Task B: Step 5 Test Retest Outcomes

<i>Current Screen View: “Order Schedule”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Again, Yes. The user is aware that they have the option to further refine the order results by looking for precise order cost ranges, orders made by specific staff or orders that have comments appended.
Can the user see a control to do that?	Yes	Yes, several. There are several price spinners and two text field search options with accompanying labels.
Is it obvious how the control works?	Yes	This is now clear owing to the addition of a stepper rather than a spinner. This now provides greater accuracy.
Does the user understand the feedback they have been given?	Yes	Yes, fields appear with selected contents of the search, or the monetary value from the spinner.

6. Search order using the applied filters

Table 36: Task B: Step 6 Test Outcomes

<i>Current Screen View: “Order Schedule”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, they want to search having selected the desired filters.

Can the user see a control to do that?	Yes	Yes, a large button at the bottom of filter options reads <i>"Search Orders"</i> . This is clearly the control to perform this action.
Is it obvious how the control works?	Yes	Yes, clearly outlined and widely familiar.
Does the user understand the feedback they have been given?	Yes	Yes, <i>"Search Results"</i> are now presented to the user. This has clearly performed the search action that the user wanted.

7. Find the correct job from the results list returned

Table 37: Task B: Step 7 Test Outcomes

<i>Current Screen View: "View Orders Search Results"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, the user wishes to search for the correct order record from the returned list of orders matching the criteria
Can the user see a control to do that?	Yes	Yes, they can. A large button showing some of the main details is shown. It is large and rectangular so is unmistakable.
Is it obvious how the control works?	Yes	Yes, users will be familiar with this control.
Does the user understand the feedback they have been given?	Yes	They have been taken to the detailed view of the order so can clearly see that this action has taken them into the order.

8. Amend the appropriate fields to update the orders information

Table 38: Task B: Step 8 Test Outcomes

<i>Current Screen View: "Order Details"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, they now wish to edit the information that they can see for this record.
Can the user see a control to do that?	Yes	Yes, a large <i>"Edit Order"</i> button can be seen at the bottom of the page screen via scroll. The fact the scroll is visible is

		enough to tell the user that there is more information and further options are available below the current screen position.
Is it obvious how the control works?	Yes	Yes, it's a large rectangular button with clear text and an icon. It is clearly a button.
Does the user understand the feedback they have been given?	Yes	Yes, the form has now changed with options to edit information. Text boxes are now showing as editable as well as order comments. Some fields are now clickable opening spinner controls.

9. Save and close the order record

Table 39: Task B: Step 9 Test Outcomes

<i>Current Screen View: "Order Details"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, a user understands that they need to commit the changes having made alterations. Users are aware they need to save what they have done.
Can the user see a control to do that?	Yes	A button is shown that shows "Save Record" – clearly showing that the information now present can be changed.
Is it obvious how the control works?	Yes	Yes, it's a button that clearly states what it will do. Perfectly obvious.
Does the user understand the feedback they have been given?	Yes	They have returned to the form in its previous view without the edit options. The updated information is being shown so it is clear that the updates made have been saved.

10. Sign out of the application

Table 40: Task B: Step 10 Test Retest Outcomes

<i>Current Screen View: "Job Details"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	This is now clear ... Each time the app launches the user is reminded that they must actively sign out before they close the application. This is a security enforcement rule.
	Yes	There is a "Sign Out" button visible at the bottom right of the screen. This looks

Can the user see a control to do that?		like a way to sign out of an account. The accompanying icon helps make this its purpose clear.
Is it obvious how the control works?	Yes	It's a button, so is a widely familiar control.
Does the user understand the feedback they have been given?	Yes	They have returned to the 'Sign In' screen and can see an option to sign into the app. They are aware that this means they have successfully signed out.

10.6.3 Walkthrough C: Create a new Maintenance Job

1. Sign in with “Staff ID” and “Password”

Table 41: Task C: Step 1 Test Outcomes

<i>Current Screen View: “Login”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, users are aware that they need to use their identification and authentication to sign themselves into the app.
Can the user see a control to do that?	Yes	The boxes for “Staff ID” (identification) and “Password” (authentication) are clearly obvious. There is also a large “Sign In” button.
Is it obvious how the control works?	Yes	There is one button and two text fields that when touched bring up a keyboard for text entry.
Does the user understand the feedback they have been given?	Yes	Having entered the correct details, the following screen shows a two-pane window with menu items on the left if the details are correct. If the attempt was unsuccessful, a clear pop-up shows the failed attempt.

2. Select the “View Power car Schedule” menu pane

Table 42: Task C: Step 2 Test Outcomes

<i>Current Screen View: “Main Menu”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, they wish to firstly view the state of the power car schedule to check for any clashes. Users know that they want to find the schedule.
Can the user see a control to do that?	Yes	Yes, a large button in the left-hand list reads “View Power car Schedule”. This looks right.
Is it obvious how the control works?	Yes	Yes, it’s a button with clear text. There should be no confusion.
Does the user understand the feedback they have been given?	Yes	The right-hand pane now shows refined options to view the work schedule. Clearly this action has seen the desired result.

3. View all the power car jobs to view current schedule capacity

Table 43: Task C: Step 4 Test Outcomes

<i>Current Screen View: "Power cars"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, they still want to show the results. They might wish to refine the results returned using controls on this page to see jobs between a certain time range.
Can the user see a control to do that?	Yes	Yes, there is a visible button to view all as well as various filters.
Is it obvious how the control works?	Yes	Yes, clearly there are buttons and date pickers that have annotated labels. No controls here that are confusing.
Does the user understand the feedback they have been given?	Yes	Yes, having selected "Search Jobs" or "View All Jobs" the results list is shown. This allows staff to check for schedule capacity before adding in a new job.

4. Go to the "Add New Record" screen in its separate menu

Table 44: Task C: Step 4 Test Outcomes

<i>Current Screen View: "Power cars"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Having checked the existing jobs, they wish to add a new job into the schedule. To do this they know they want to create a new job in the system.
Can the user see a control to do that?	Yes	Yes.... <u>But possibly not first time.</u> It might take a few seconds for the user to see no option to add a record on the right-hand pane. When looking over to the menu pane, and downwards, "Add New Record" is a likely control. This can be learnt quickly after first use and therefore does not constitute an unobvious interaction ("?", "No").
Is it obvious how the control works?	Yes	Yes, it is clearly a button and is easily understood.
	Yes	Yes, a completely new right-hand pane is now being shown reading "Create New Record".

Does the user understand the feedback they have been given?		
--	--	--

5. Specify the type of record to be added is a maintenance job

Table 45: Task C: Step 5 Test **Retest** Outcomes

<i>Current Screen View: "Add New Record"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	They still want to create this new maintenance job in the system.
Can the user see a control to do that?	Yes	They can see a spinner field underneath the page heading, but other than this it is blank. A heading suggesting that the user must select the type of record is clear and obvious, prompting the user to select the desired type.
Is it obvious how the control works?	Yes	It is a combo-box type spinner and is widely used in many mobile interfaces. It should be clear how to use this control.
Does the user understand the feedback they have been given?	Yes	Having selected "Create New Maintenance Job" a range of blank fields and accompanying controls have appeared. Clearly selecting an option from this dropdown has had some effect.

6. Create the new maintenance record with desired information

Table 46: Task C: Step 6 Test Outcomes

<i>Current Screen View: "Add New Record"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	They can see a blank form and want to complete the appropriate fields with new information.
Can the user see a control to do that?	Yes, many.	There are numerous fields that the user can see to add in various pieces of information regarding the new job.
Is it obvious how the control works?	Yes	Yes, many are text-based that involve a simple text box or list items. Date or time-based information is being run off spinner pickers. <i>A user is also likely to learn how to use each field because the</i>

		<i>format of data entry is the same as editing existing records.</i>
Does the user understand the feedback they have been given?	Yes	The fields complete when text is added, lists are created when date picker values are selected.

7. Save the newly created job record

Table 47: Task C: Step 7 Test Outcomes

<i>Current Screen View: "Add New Record"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	They want to save the new record that they have just completed.
Can the user see a control to do that?	Yes	A "Save Record" button is located at the bottom of screen, visible when the user scrolls down to complete the record information.
Is it obvious how the control works?	Yes	It is a button so it is clear that this should be tapped to save the new record.
Does the user understand the feedback they have been given?	Yes	The record has been saved and the state of the form has changed. It is no longer editable so the process of saving the form state has clearly been successfully.

8. Sign out of the application

Table 48: Task C: Step 8 Test Outcomes

<i>Current Screen View: "Job Details"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	This is now clear... Each time the app launches the user is reminded that they must actively sign out before they close the application. This is a security enforcement rule.
Can the user see a control to do that?	Yes	There is a "Sign Out" button visible at the bottom right of the screen. This looks like a way to sign out of an account. The accompanying icon helps make its purpose clear.
Is it obvious how the control works?	Yes	It's a button, so is a widely familiar control.

Does the user understand the feedback they have been given?	Yes	They have returned to the 'Sign In' screen and can see an option to sign into the app. They are aware that this means they have successfully signed out.

---End of Cognitive Walkthrough Scripts---

10.7 Wireframe Testing: Usability Test Scripts

Full usability testing interviews, during which these scripts were completed, can be seen in the attached project files.

10.7.1 Participant A

☐ Web browser should be open to Google or some other “neutral” page

☐ Give them a recording permission form and a pen
☐ While they sign it, **START the SCREEN RECORDER**

1. Introduction

Hi. My name is Henry, and I’m going to be walking you through this session today.

Before we begin, I have some information for you, and I’m going to read it aloud to make sure that I cover everything.

You probably already have a good idea of why I’ve asked you here but let me go over it again briefly. I’m asking people to try using an app that I have designed so I can see whether it works as intended. The session should take at most around 20-25 minutes.

The first thing I want to make clear right away is that I’m testing the product, not you. You can’t do anything wrong here. In fact, this is probably the one place today where you don’t have to worry about making mistakes.

As you use the app, I’m going to ask you as much as possible to try to think out loud: to say what you’re looking at, what you’re trying to do, and what you’re thinking. This will be a big help to me.

Also, please don’t worry that you’re going to hurt my feelings. I’m doing this to improve the app, so I need to hear your honest reactions.

If you have any questions as we go along, just ask them. I may not be able to answer them right away, since I'm interested in how people do when they don't have someone sitting next to them to help. But if you still have any questions when we're done, I'll try to answer them then. And if you need to take a break at any point, just let me know.

You may have noticed the microphone. With your permission, we're going to record what happens on the screen and our conversation. The recording will only be used to help me figure out how to improve the app, and it won't be seen by anyone except the people working on this project. And it helps me, because I don't have to take as many notes.

If you would, I'm going to ask you to sign a simple permission form for us. It just says that we have your permission to record you, and that the recording will only be seen by the people working on the project.

--- Signature ---

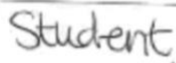
A rectangular box containing a handwritten signature in black ink.

2. Understanding the Interviewee

Do you have any questions so far?

OK. Before we look at the site, I'd like to ask you just a few quick questions.

Firstly, what's your occupation?

A rectangular box containing the handwritten word "Student." in black ink.

Now, roughly how many hours a week altogether—just a ballpark estimate— would you say you spend using the Internet, including Web browsing and email, at work and at home?

8 hours a day including uni work.
Around a week that is 56 or so hours.

And what's the split between email and browsing—a rough percentage?

5% email
95% work/study browsing.

What kinds of sites are you looking at when you browse the Web?

Uni/Quick links.
Google, YouTube. Work/study specific.

Do you have any favourite Web sites?

Amazon. YouTube as well.
FB Messenger, SKYScanner. (flight app).

OK, great. We're done with the questions, and we can start looking at things.

3. Home Page

☐ **Click on the bookmark for the site's Home page.**

First, I'm going to ask you to look at this page and tell me what you make of it: what strikes you about it, whose site you think it is, what you can do here, and what it's for. Just look around and do a little narrative.

You can scroll if you want/need to, but don't click on anything yet.

Not quirky but it doesn't look too formal.

More consumer, intise the user. Colour is less professional. Logo is professional.

☐ **Allow this to continue for three or four minutes, at most.**

4. User Tasks

Thanks. Now I'm going to ask you to try doing some specific tasks. I'm going to read each one aloud, but I'll repeat each task throughout whenever you wish.

And again, as much as possible, it will help us if you can try to think out loud as you go along.

* Give brief about App background - 2 mins about Scenario. *

- ☐ Hand the participant the first scenario, and read it aloud.
- ☐ Allow the user to proceed until you don't feel like it's producing any value or the user becomes very frustrated.
- ☐ Repeat for each task or until time runs out.

4.1. Task 1

"Mark a maintenance job as complete by filling out any blank fields in an existing job record"

Additional Task Information (if required):

- Find a record of a power car job in the app
- Edit the existing record found
- Mark the work as complete
- Save the changes
- Sign out of the application

Signed In as before.
 Gone to "View Power car" schedule.
 Gone straight to "View all".
 Clicked on shown job. "Edit." (not on screen)
 Then went back to "Sign In." to sign out.
"Edit an existing Component order that has been placed to add extra items to it"

Additional Task Information (if required):

- Find an existing order record in the app
- Edit the existing record found
- Add at least one new component to the order

- Save the changes to the order

~~Sign in~~ "Signed In" to begin with.
 Then went to "View Orders". Clicked on
 "View All" and then on an order. (one from list)
 "Edit Order" to add item and then "Save".
 "Signed Out" to return to home page.

4.3. Task 3

"Create a new maintenance job for a power car in the power car schedule where there is sufficient space by first checking the existing schedule entries"

was looking for 'home' / 'back'.

Additional Task Information (if required):

- Find the existing power car schedule records
- Then, create a new power car job in the schedule
- Add desired settings, comments and timescales as you wish
- Save the changes to the record
- Sign out of the application

"sign in" to get to main.
 "View Power car schedule" to bring up
 Search list.
 "Search jobs" brings up list of work.
 "Correct spinner" then went down to "Create
 Do you have any questions for me, now that we're done or any other Record
 comments you would like to raise?"

* Participant requested to explore for a few seconds.*

Looks well structured. "Seems fairly fluid."

✓ "Informative but busy" "Quite concise layout."

Thank you for your time and participation, that was very helpful.

✓ "Would like 5 mins before to view and play before tasks started." "Everything is well labelled."

10.7.2 Participant B

☐ Web browser should be open to Google or some other “neutral” page

☐ Give them a recording permission form and a pen

☐ While they sign it, **START the SCREEN RECORDER**

1. Introduction

Hi. My name is Henry, and I’m going to be walking you through this session today.

Before we begin, I have some information for you, and I’m going to read it aloud to make sure that I cover everything.

You probably already have a good idea of why I’ve asked you here but let me go over it again briefly. I’m asking people to try using an app that I have designed so I can see whether it works as intended. The session should take at most around 20-25 minutes.

The first thing I want to make clear right away is that I’m testing the product, not you. You can’t do anything wrong here. In fact, this is probably the one place today where you don’t have to worry about making mistakes.

As you use the app, I’m going to ask you as much as possible to try to think out loud: to say what you’re looking at, what you’re trying to do, and what you’re thinking. This will be a big help to me.

Also, please don’t worry that you’re going to hurt my feelings. I’m doing this to improve the app, so I need to hear your honest reactions.

If you have any questions as we go along, just ask them. I may not be able to answer them right away, since I'm interested in how people do when they don't have someone sitting next to them to help. But if you still have any questions when we're done, I'll try to answer them then. And if you need to take a break at any point, just let me know.

You may have noticed the microphone. With your permission, we're going to record what happens on the screen and our conversation. The recording will only be used to help me figure out how to improve the app, and it won't be seen by anyone except the people working on this project. And it helps me, because I don't have to take as many notes.

If you would, I'm going to ask you to sign a simple permission form for us. It just says that we have your permission to record you, and that the recording will only be seen by the people working on the project.

--- Signature ---

A handwritten signature in black ink, appearing to be 'G. Cole', is written inside a rectangular box.

2. Understanding the Interviewee

Do you have any questions so far?

OK. Before we look at the site, I'd like to ask you just a few quick questions.

Firstly, what's your occupation?

A handwritten response in black ink, 'Work at Ikea in Exeter.', is written inside a rectangular box.

Now, roughly how many hours a week altogether—just a ballpark estimate— would you say you spend using the Internet, including Web browsing and email, at work and at home?

21+ hours per week using browsing + email.

And what's the split between email and browsing—a rough percentage?

95% browsing and 5% Email.

Similar for smartphones, more email than browsers.

What kinds of sites are you looking at when you browse the Web?

Video streaming sites on YouTube.
Clothes/technology apps/sites.

Do you have any favourite Web sites?

YouTube is a favourite.
Apart from that fairly spread.

OK, great. We're done with the questions, and we can start looking at things.

3. Home Page

☐ **Click on the bookmark for the site's Home page.**

First, I'm going to ask you to look at this page and tell me what you make of it: what strikes you about it, whose site you think it is, what you can do here, and what it's for. Just look around and do a little narrative.

You can scroll if you want/need to, but don't click on anything yet.

Opening page, clearly opening of the app and who its for.
Abbreviation is obvious but doesn't give obvious background. No way out but clearly a way in / forwards via Sign In.

☐ **Allow this to continue for three or four minutes, at most.**

4. User Tasks

Thanks. Now I'm going to ask you to try doing some specific tasks. I'm going to read each one aloud, but I'll repeat each task throughout whenever you wish.

And again, as much as possible, it will help us if you can try to think out loud as you go along.

- ☐ Hand the participant the first scenario, and read it aloud.
- ☐ Allow the user to proceed until you don't feel like it's producing any value or the user becomes very frustrated.
- ☐ Repeat for each task or until time runs out.

4.1. Task 1

"Mark a maintenance job as complete by filling out any blank fields in an existing job record"

Additional Task Information (if required):

- Find a record of a power car job in the app
- Edit the existing record found
- Mark the work as complete
- Save the changes
- Sign out of the application

Added in details and signed in to go to "Home Page".

Selected "View Power car schedule". Feel as though not in right place. Then "View all". Then gone "View Job" to expand.

"Edit an existing Component order that has been placed to add extra items to it"

Additional Task Information (if required):

- Find an existing order record in the app
- Edit the existing record found
- Add at least one new component to the order

STRUGGLING TO FIND RECORD TO EDIT.
option to edit eventually found.

- Save the changes to the order

Went straight to "view orders." Not sure about orders of this. Might possibly take short time to learn.

Then went to "View All Orders", then on an order to view its details. Then went to "~~XXXXXX~~ Edit Order."

4.3. Task 3

Would expect more detailed information here. When editing.

"Create a new maintenance job for a power car in the power car schedule where there is sufficient space by first checking the existing schedule entries"

Additional Task Information (if required):

- Find the existing power car schedule records
- Then, create a new power car job in the schedule
- Add desired settings, comments and timescales as you wish
- Save the changes to the record
- Sign out of the application

* Identified "Sign Out" as "home".
"Signed In" → up to "Power car schedule", then "view all" to see what currently there is. "Add New Record", "dropdown".
Why is there a switch to show text box held.

Do you have any questions for me, now that we're done or any other comments you would like to raise?

Audio queues. Haptic feedback. Would be nice to have clear feedback when filling out new forms, saving, deleting.

Thank you for your time and participation, that was very helpful.

10.7.3 Participant C

☐ **Web browser should be open to Google or some other “neutral” page**

☐ **Give them a recording permission form and a pen**

☐ **While they sign it, START the SCREEN RECORDER**

1. Introduction

Hi. My name is Henry, and I’m going to be walking you through this session today.

Before we begin, I have some information for you, and I’m going to read it aloud to make sure that I cover everything.

You probably already have a good idea of why I’ve asked you here but let me go over it again briefly. I’m asking people to try using an app that I have designed so I can see whether it works as intended. The session should take at most around 20-25 minutes.

The first thing I want to make clear right away is that I’m testing the product, not you. You can’t do anything wrong here. In fact, this is probably the one place today where you don’t have to worry about making mistakes.

As you use the app, I’m going to ask you as much as possible to try to think out loud: to say what you’re looking at, what you’re trying to do, and what you’re thinking. This will be a big help to me.

Also, please don’t worry that you’re going to hurt my feelings. I’m doing this to improve the app, so I need to hear your honest reactions.

If you have any questions as we go along, just ask them. I may not be able to answer them right away, since I'm interested in how people do when they don't have someone sitting next to them to help. But if you still have any questions when we're done, I'll try to answer them then. And if you need to take a break at any point, just let me know.

You may have noticed the microphone. With your permission, we're going to record what happens on the screen and our conversation. The recording will only be used to help me figure out how to improve the app, and it won't be seen by anyone except the people working on this project. And it helps me, because I don't have to take as many notes.

If you would, I'm going to ask you to sign a simple permission form for us. It just says that we have your permission to record you, and that the recording will only be seen by the people working on the project.

--- Signature ---

A handwritten signature in black ink, consisting of a stylized 'S' followed by a horizontal line.

2. Understanding the Interviewee

Do you have any questions so far?

OK. Before we look at the site, I'd like to ask you just a few quick questions.

Firstly, what's your occupation?

PhD Student and part-time lecturer.

Now, roughly how many hours a week altogether—just a ballpark estimate— would you say you spend using the Internet, including Web browsing and email, at work and at home?

30-40 hrs per week all in.

And what's the split between email and browsing—a rough percentage?

10% email so 90% is browsing for web.
Mobile is very broad usage.

What kinds of sites are you looking at when you browse the Web?

Email, browsing, video, games and social media and online shopping.

Do you have any favourite Web sites?

Gmail, media (Netflix, Spotify, play-store) and Chebrive. Social Media.

OK, great. We're done with the questions, and we can start looking at things.

3. Home Page

☐ **Click on the bookmark for the site's Home page.**

First, I'm going to ask you to look at this page and tell me what you make of it: what strikes you about it, whose site you think it is, what you can do here, and what it's for. Just look around and do a little narrative.

You can scroll if you want/need to, but don't click on anything yet.

Looks like Staff portal. Straightforward
Sign in for staff. Knew it would be
GWR. Not for customers, staff.

☐ **Allow this to continue for three or four minutes, at most.**

4. User Tasks

Thanks. Now I'm going to ask you to try doing some specific tasks. I'm going to read each one aloud, but I'll repeat each task throughout whenever you wish.

And again, as much as possible, it will help us if you can try to think out loud as you go along.

- ☐ Hand the participant the first scenario, and read it aloud.
- ☐ Allow the user to proceed until you don't feel like it's producing any value or the user becomes very frustrated.
- ☐ Repeat for each task or until time runs out.

4.1. Task 1

"Mark a maintenance job as complete by filling out any blank fields in an existing job record"

Additional Task Information (if required):

- Find a record of a power car job in the app
- Edit the existing record found
- Mark the work as complete
- Save the changes
- Sign out of the application

Knew had to "sign in" by entering details in both boxes.

"View Powercar schedule." shows the results Search page. Clearly identified. Clearly stated purpose

"Search Results" shows results and then through "Edit an existing Component order that has been placed to add extra to job. items to it"

Additional Task Information (if required):

- Find an existing order record in the app
- Edit the existing record found
- Add at least one new component to the order

Numerous fields clearly identified.
No issues occurred all.

- Save the changes to the order

Went straight to "View Orders" tab. Clearly identify the form controls and purpose. Clearly sees search parameters and results. Perfectly sees that ~~reverse~~ would need reverse lookup.

4.3. Task 3

"Create a new maintenance job for a power car in the power car schedule where there is sufficient space by first checking the existing schedule entries"

Additional Task Information (if required):

- Find the existing power car schedule records
- Then, create a new power car job in the schedule
- Add desired settings, comments and timescales as you wish
- Save the changes to the record
- Sign out of the application

"View Power car schedule" to view planned dates. Goes to "Add New Record" then not sure about how this works with searching an order that exists.

Do you have any questions for me, now that we're done or any other comments you would like to raise?

Self-explanatory, works as it says on the tin.
Need to look at how Maintenance component links with an order-based system as to where they are going.

Thank you for your time and participation, that was very helpful.

"Stores team status" + "allocated"

10.8 Wireframe Testing: System Usability Scale Scripts

10.8.1 Participant A

Participant ID: USER1 Site: GWR: BALSAMIQ Date: 15/04/19

System Usability Scale

Instructions: For each of the following statements, mark one box that best describes your reactions to the website *today*.

	Strongly Disagree				Strongly Agree
1. I think that I would like to use this website frequently.	☐	☐	☒	☐	☐
2. I found this website unnecessarily complex.	☐	☒	☐	☐	☐
3. I thought this website was easy to use.	☐	☐	☐	☒	☐
4. I think that I would need assistance to be able to use this website.	☐	☒	☐	☐	☐
5. I found the various functions in this website were well integrated.	☐	☐	☐	☐	☒
6. I thought there was too much inconsistency in this website.	☐	☒	☐	☐	☐
7. I would imagine that most people would learn to use this website very quickly.	☐	☐	☒	☐	☐
8. I found this website very cumbersome/awkward to use.	☐	☐	☒	☐	☐
9. I felt very confident using this website.	☐	☐	☒	☐	☐
10. I needed to learn a lot of things before I could get going with this website.	☐	☒	☐	☐	☐

Please provide any comments about this website:

10.8.2 Participant B

Participant ID: USER 2 Site: GWR: BALSAMIC Date: 16/04/19

System Usability Scale

Instructions: For each of the following statements, mark one box that best describes your reactions to the website *today*.

	Strongly Disagree				Strongly Agree
1. I think that I would like to use this website frequently.	☐	☐	☒	☐	☐
2. I found this website unnecessarily complex.	☐	☒	☐	☐	☐
3. I thought this website was easy to use.	☐	☐	☐	☒	☐
4. I think that I would need assistance to be able to use this website.	☐	☐	☐	☒	☐
5. I found the various functions in this website were well integrated.	☐	☒	☐	☐	☐
6. I thought there was too much inconsistency in this website.	☐	☒	☐	☐	☐
7. I would imagine that most people would learn to use this website very quickly.	☐	☐	☐	☒	☐
8. I found this website very cumbersome/awkward to use.	☐	☐	☒	☐	☐
9. I felt very confident using this website.	☐	☒	☐	☐	☐
10. I needed to learn a lot of things before I could get going with this website.	☐	☐	☒	☐	☐

Please provide any comments about this website:

lexis and dialect, hard to put yourself in their shoes.

Clear direction on menu components but more information on how to use each page.

Participant ID: USER 3 Site: GWIR : BALSAMIC Date: / /

System Usability Scale

Instructions: For each of the following statements, mark one box that best describes your reactions to the website *today*.

	Strongly Disagree				Strongly Agree
1. I think that I would like to use this website frequently.	☐	☐	☐	☒	☐
2. I found this website unnecessarily complex.	☒	☐	☐	☐	☐
3. I thought this website was easy to use.	☐	☐	☐	☐	☒
4. I think that I would need assistance to be able to use this website.	☐	☒	☐	☐	☐
5. I found the various functions in this website were well integrated.	☐	☒	☐	☒	☐
6. I thought there was too much inconsistency in this website.	☒	☐	☐	☐	☐
7. I would imagine that most people would learn to use this website very quickly.	☐	☐	☐	☐	☒
8. I found this website very cumbersome/awkward to use.	☒	☐	☐	☐	☐
9. I felt very confident using this website.	☐	☐	☐	☒	☐
10. I needed to learn a lot of things before I could get going with this website.	☒	☐	☐	☐	☐

Please provide any comments about this website:

This questionnaire is based on the System Usability Scale (SUS), which was developed by John Brooke while working at Digital Equipment Corporation. © Digital Equipment Corporation, 1986.

10.8.4 Participant A SUS Score Table

Table 49: User A SUS scoresheet

Question	Calculated Question Score
1	2
2	3
3	3
4	3
5	4
6	3
7	2
8	2
9	2
10	3

Total Score for User A: **27**

Calculated SUS score for User A: **27 * 2.5 = 67.5**

10.8.5 Participant B SUS Score Table

Table 50: User B SUS scoresheet

Question	Calculated Question Score
1	2
2	3
3	3
4	1
5	1
6	3
7	3
8	2
9	1
10	2

Total Score for User A: **21**

Calculated SUS score for User A: **21 * 2.5 = 52.5**

10.8.6 Participant C SUS Score Table

Table 51: User C SUS scoresheet

Question	Calculated Question Score
1	3
2	4
3	4
4	3
5	3
6	4
7	4
8	4
9	3
10	4

Total Score for User A: **36**

Calculated SUS score for User A: **36 * 2.5 = 90**

10.9 User Testing: Formal Test Documentation

10.9.1 Participant A Test Reports

Table 52: Test Report Legend

Legend	Key	Notes
	Positive Finding	<i>"Works well. Approach is recommendable"</i>
	Minor Problem	<i>"Minor dissatisfaction; noticeable delays; or superficial difficulties"</i>
	Major Problem	<i>"Substantial delays; or moderate dissatisfaction"</i>
	Functional Problem	<i>"Bug. The product does not work as specified"</i>
	Good Idea	<i>"A suggestion from the test participant or the moderator that could lead to a significant improvement of the user experience"</i>

Table 53: Participant A: Initial Thoughts

Test Stage	Test Outcome(s)	Test Action(s)
Initial Thoughts of Application	1. The colour wasn't too professional but did think the logo looked good. 2. The "Start" screen doesn't look too formal. Quote <i>"wouldn't be a business app"</i> because of the colour, the logo did make it look more professional.	

Table 54: Participant A: Scenario 1

Test Stage	Test Outcome(s)	Test Action(s)
Scenario 1	1. Signed in fine and understood that he had to add in details before clicking on "Sign in". 2. Quickly went to "View Power car Schedule" pane on the left. 3. Went straight to "View All" jobs after looking at the date filters on the search records view on this page. 4. Clearly identified a record to bring up its details. Took him a little while to decide to "Edit the job". 5. Wanted to go to this screen but it was not part of the initial design sets. Tester then explains what and how this would happen (some checking by me to see if the designs are in the report or anywhere else) 6. Next asked the user to go "home" before starting scenario 2. On the power car maintenance record view, the user quickly went to the back button to return	     

	to “<i>Search Results</i>” page. Looked and acted as if they were very unsure as to how they could get from “<i>Search Results</i>” back a “<i>Home</i>” or “<i>Start</i>” page. 7. Eventually clicked on “<i>View Power car Schedule</i>” which brought up the “<i>Power car</i>” search page but still looked as if they were unsatisfied by this. Eventually they looked to go to “<i>Sign In</i>”. (though this is different to the initial “<i>home page</i>” a user sees when signing in)	 
--	---	--

Table 55: Participant A: Scenario 2

Test Stage	Test Outcome(s)	Test Action(s)
Scenario 2	1. “<i>Sign In</i>” was fast. 2. User’s initial reaction was to go to the “<i>View Orders</i>” page. 3. Then went to “<i>View All Orders</i>” without scrolling down (again not using “<i>Search Orders</i>”). 4. Clearly then clicked on a result that could be found in the list. 5. Was a little hesitant but clicked on go to “<i>Edit Order</i>”. Then again, the interviewer explains to the user that the next screen is not present but that controls would become editable alongside an option to save any changes.	    

Table 56: Participant A: Scenario 3

Test Stage	Test Outcome(s)	Test Action(s)
Scenario 3	1. Started back at the login screen and signed in quickly. 2. Went straight to “<i>View Power car Schedule</i>”. 3. Then went to “<i>View all Jobs</i>” before the interviewer makes the interviewee aware of the option that would be present to fill out some date filters before searching those specific parameters. 4. User then imagined they had added in a date range before going “<i>Search Jobs</i>” to bring up the same “<i>Search Results</i>” view. 5. Initial reaction was to go to “<i>Add New Record</i>” before showing the field that shows the type of record that they wish to create. 6. Eventually clicking “<i>Create Record</i>” to go back to the home page.	     

Table 57: Participant A: Open Discussion

Test Stage	Test Outcome(s)	Test Action(s)
Open Discussion	Participant wished to have a further explore of the system and said the system felt <i>"pretty well structured"</i>. Wanted to ask further questions about Scenario 1 and the interviewer explains that the screen is not there but that controls and <i>"Save"</i> would be used. The interviewee explains that they felt disorientated by the extra annotations that show various control states. Good for the design and informative but maybe made completing tasks a little slower. Felt the layout was overall <i>"quite concise"</i>.	 

10.9.2 Participant B Test Reports

Table 58: Participant B: Initial Thoughts

Test Stage	Test Outcome(s)	Test Action(s)
Initial Thoughts of Application	Was clear who the app is for and it was clear that the user would have to sign in. No obvious way of going out from here apart from signing in.	

Table 59: Participant B: Scenario 1

Test Stage	Test Outcome(s)	Test Action(s)
Scenario 1	1. Clearly knew that they had to add in their details before clicking "<i>Sign In</i>". 2. Was a little hesitant but went to "<i>View Power car Schedule</i>". On the "<i>Power car</i>" page (with record search parameters) they were "<i>not sure they were in the right place at all</i>" before the user was willing to go on "<i>View All Jobs</i>". Did think that the wording on this page was interesting in places and unfamiliar. 3. Clicked on the top Job record to bring up its detailed view. Did not click on "<i>Edit Job</i>" here because they thought that they could edit the information fields currently being displayed. It took them a while to realise that they were looking for an option to edit what they were currently seeing. They were still not completely sure they were in the right place despite this.	  

Table 60: Participant B: Scenario 2

Test Stage	Test Outcome(s)	Test Action(s)
Scenario 2	- Instantly identified the “<i>View Orders</i>” menu pane on the left without looking anywhere else (such as for a “<i>Back</i>” or “<i>Home</i>” option) - Not sure about “<i>Order Schedule</i>” filter options - thought that it might influence the speed of user learning. - Then went “<i>View All Orders</i>” and then clicking on an order to allow the order to	  

10.9.3 Participant C Test Reports

Table 63: Participant C: Initial Thoughts

Test Stage	Test Outcome(s)	Test Action(s)
Initial Thoughts of Application	First impression is that it looks like a staff portal system using a straightforward login. Knows GWR as Great Western Railway. Doesn't look like a customer-related application.	

Table 64: Participant C: Scenario 1

Test Stage	Test Outcome(s)	Test Action(s)
Scenario 1	1. First thing is what do I need to login for both the password and the staff ID. Does it allow passwords to be remembered? Clearly understands this.	
	2. First thing is orientation has changed to landscape. First thoughts are to go to “View Power car Schedule”, bringing up the “Power Car” search page. Understands that these controls are search parameters and talks the interviewer through the different options, such as staff, site or vehicle number. Can clearly see that this is a search criteria screen. <i>This participant did quickly understand that the “View All Jobs” was a lazy option that doesn’t perform a filter.</i> For this task user clicks on “View All Jobs”.	
	3. Presented with job lists, clearly identifies that they have the option to return to the search criteria page. User is presented with “Job Details” page.	
	4. Scrolls down to view all the record fields and talks through the different fields. Says that they can see a control that contains component ordering scheduling. Highlights the delete option for the job. Clearly identifies the purpose and contents of the page.	

Table 65: Participant C: Scenario 2

Test Stage	Test Outcome(s)	Test Action(s)
Scenario 2	1. User went straight to the “View Orders” page without looking to go “Home”.	
	2. User clearly identifies the controls and options by which to refine the order records	

	presented to the user. Clicks on <i>“View All Orders”</i> to see all results. 3. Clicks on the order in the results list that provides <i>“quite a good bit of information”</i>. Cancellation information and delivery information. Option to add comments or speak to the suppliers controls clearly identified. Did state that this page didn’t link a job to a vehicle – participant identified the comments box as an option to add this information in. <i>“If this order came in how you would know what vehicle this was meant for?”</i> <i>“Would there be a correlation between the two?”</i> <i>“Almost a reverse lookup”</i>	   
--	--	---

Table 66: Participant C: Scenario 3

Test Stage	Test Outcome(s)	Test Action(s)
Scenario 3	- Went straight to <i>“View Power car Schedule”</i> and then to <i>“View All Jobs”</i> to view what was planned. - Having looked at the search results the user went to the left pane and <i>“Add New Record”</i>. Clearly identified the option at the top that must be selected. Not entirely sure about how the first menu selection works having selected an option at the top of the screen. Struggled to make sense of the <i>“Order ID”</i> field to place an order against a job. - Identifies the option to <i>“Create Record”</i> or <i>“Cancel”</i>. Presses <i>“Create Record”</i> to take the user back to <i>“Main Menu”</i>.	   

Table 67: Participant C: Open Discussion

Test Stage	Test Outcome(s)	Test Action(s)
Open Discussion	<i>“Fairly straightforward, fairly self-explanatory and does what it says on the tin”</i> . Only suggestions relating to reverse lookup where order records show the order destination. Currently the job records show the orders that are destined for them so this would involve a reverse lookup option.	

10.10 Prototype Testing: Cognitive Walkthrough Evaluation

10.10.1 Walkthrough A: Marking a Maintenance Job Complete

1. Open the Application

Table 68: Task A: Step 1 Test Outcomes

Current Screen View: <i>*Users device interface*</i>		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	Yes	User knows they want to go to the app
Can the user see a control to do that?	Yes	Yes, the app will be shown on the user's device for them to open
Is it obvious how the control works?	Yes	It is <i>highly</i> likely that this will be a button on the user's mobile device, however it could depend on the device used.
Does the user understand the feedback they have been given?	Yes	The application opens on the device. Yes, they understand the result of this action

2. Sign In using assigned Appenate credentials

Table 69: Task A: Step 2 Test Outcomes

Current Screen View: <i>"Sign In"</i>		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	Yes	Yes, they are aware that they need to sign in to gain access to the contents of the application by entering their correct details.
Can the user see a control to do that?	Yes, several	Yes, there is an "Email Address" box prompting the user to enter their email address. Having entered an email and clicking "Next" a "Password" field appears. User can then enter their account password.
Is it obvious how the control works?	Yes	Yes, they have to enter their authentication and validation details before clicking "Sign In".

Does the user understand the feedback they have been given?	Yes	Yes, successful sign-in shows the apps main menu, incorrect sign-in asks the user to try again.
--	------------	---

3. Select the “Vehicle Maintenance” menu item

Table 70: Task A: Step 3 Test Outcomes

<i>Current Screen View: “Home”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, they want to update an existing job in the system to complete empty fields and sign off the work.
Can the user see a control to do that?	Yes	Yes. In order to mark something as complete they need to view the work jobs. A button showing “Vehicle Maintenance” is the most likely option on the “Home Screen”.
Is it obvious how the control works?	Yes	Yes, it’s a button that the user will know that they have to tap.
Does the user understand the feedback they have been given?	Yes	A new screen is displayed that is clearly different to the “Home” screen.

4. Select the correct Maintenance Job ID

Table 71: Task A: Step 4 Test Outcomes

<i>Current Screen View: “Vehicle Maintenance”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	A user wants to select a job that already exists in the system. To do this they need to find the one they want.
Can the user see a control to do that?	Yes	The field at the top of the page says “Select a Maintenance Job”. Leaving it blank informs the user that they will be creating a new record so they are aware they need to find the job
Is it obvious how the control works?	Yes	There is an empty box with a dropdown arrow on the right-hand side. User

		would be expected to tap on this arrow to bring up the spinner wheel.
Does the user understand the feedback they have been given?	Yes	Yes, a wheel appears at the bottom of the screen with the caption "Select a Job ID". The user can see that they should scroll through to select one, during which the other fields populate with record data.

5. Update the various screen fields as desired involving scroll

Table 72: Task A: Step 5 Test **Retest** Outcomes

<i>Current Screen View: "Vehicle Maintenance"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, they still wish to make changes to the record that they have now selected.
Can the user see a control to do that?	Yes, many.	Yes, they can see dropdown icons, date fields and text boxes which provide them with controls to edit the data contained in the record.
Is it obvious how the control works?	Unable to Fix	The 'DateTime' fields cannot be fixed owing to the nature of the builder. Both 16:00 and 04:00 are shown in the spinner results as: ---"29-04-2018 04:00"--- This is not due to the T-SQL database created by the user as the contents added to 'DateTime' fields are of the 24-hour type.
Does the user understand the feedback they have been given?	Yes	The action of changing any of these fields is clear and obvious to a user.

6. Sign off job in correct field

Table 73: Task A: Step 6 Test Outcomes

<i>Current Screen View: "Vehicle Maintenance"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>

Does the user know what needs to be done?	Yes	They have found the job and edited the applicable fields – they know that they wish to mark the work as complete
Can the user see a control to do that?	Yes	“Signed Off By” is the obvious field towards the bottom of the screen. The user would then understand that they must select their ID to sign off the work. The adjacent ‘DateTime’ field captures the timestamp of this action.
Is it obvious how the control works?	Yes	It’s a spinner control that the user has already interacted with to find the record. The user is expected to understand that it works in this way.
Does the user understand the feedback they have been given?	Yes	The ID field has populated with their identification number. It is anticipated that the user will understand what their action has caused.

7. Click “Save” to save updated record

Table 74: Task A: Step 7 Test Outcomes

<i>Current Screen View: “Vehicle Maintenance”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	The user knows that they want to “Save” an existing record, not create a new one.
Can the user see a control to do that?	Yes	It is anticipated that users would want to try to “Save” before “Create”. This is a button at the bottom of the form showing as green text on a grey button outline.
Is it obvious how the control works?	Yes	Yes, users will be familiar with the button control, as seen here.
Does the user understand the feedback they have been given?	Yes	Yes, a message appears at the bottom of the screen informing the user that the save was successful. The app returns to the “Home” screen. Although this action could be somewhat unexpected, this test has passed as the action of saving the changes has been performed and the user is expected to see that this has taken place.

8. Select the “Vehicle Maintenance” menu item

This step has already been performed by the user in *Step 3*. but this second use of the menu is significant in terms of the user understanding their goal. At this stage a user has completed half of the task and could be expressing different thoughts. This is an issue of less significance in *Step 3*. as the user is still working towards fulfilling the first stage of this task.

Table 75: Task A: Step 8 Test **Retest** Outcomes

<i>Current Screen View: “Home”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Limited by builder interface and erroneous upload functionality – unable to fix	The user must save and then re-enter this menu to upload because attempting to upload before saving frequently resulted in erroneous capture of the edited record. The functionality save and upload has proved ungainly and is expected to cause some user confusion.
Can the user see a control to do that?	Yes	Yes, a button reading “Vehicle Maintenance” is clearly evident.
Is it obvious how the control works?	Yes	Yes, its unmistakably a button.
Does the user understand the feedback they have been given?	Yes	Yes, they are taken to a page that is showing a completely different view. The user is likely to be aware that this has performed a precise action.

9. Select the “View Saved” option

Table 76: Task A: Step 9 Test Outcomes

<i>Current Screen View: “Home”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	If , the user has clicked again to open the Job list screen they might now understand that saving the record was not enough to upload it. They might now be aware that they must still upload the changes made to the record.
Can the user see a control to do that?	Yes	Yes, when opening the screen they were asked if they want to view saved records or make a new change to the maintenance jobs.
	Yes	It’s a button so the user will understand that they need to click on “View Saved”

Is it obvious how the control works?		
Does the user understand the feedback they have been given?	Yes	Yes, a new page is displayed that shows all the Saved changes made by users on records in this list. Clearly see the results of clicking this button.

10. Select the correct edit instance from the returned list of edit instances

Table 77: Task A: Step 10 Test Outcomes

<i>Current Screen View: “Vehicle Maintenance – Saved Entries”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	A user is expected to now want to find the record that was changed in order to push the changes back to Appenate.
Can the user see a control to do that?	On balance, Yes	Results are ordered by the timestamp and it is anticipated that a user will rapidly recognise what is being shown here.
Is it obvious how the control works?	On balance, Yes	A user might not recognise each result as a button that they can click, however it is anticipated that their first reaction would be to tap.
Does the user understand the feedback they have been given?	Yes	Yes, an option box is now being shown to the user.

11. Click “Open” to view it

Table 78: Task A: Step 11 Test Outcomes

<i>Current Screen View: “Vehicle Maintenance – Saved Entries”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	The user is aware that they don’t want to delete their edited record, so are more likely to want to view it in order to upload its changes. Less likely, but still a possibility, the user will wish to go backwards and exit from this task.
Can the user see a control to do that?	Yes	Yes, a clear button showing “Open” is displayed to the user.

Is it obvious how the control works?	Yes	Yes, it's a button and is displayed in the native OS style. This is expected to be easily understood by users.
Does the user understand the feedback they have been given?	Yes	The altered record is now displayed. The action of clicking this button is clear and obvious.

12. Click "Create" to upload the record

Table 79: Task A: Step 12 Test Outcomes

Current Screen View: "Vehicle Maintenance"		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	Yes	They still wish to upload the record that they have saved (on the premise that a user does understand that they haven't completed the task)
Can the user see a control to do that?	Yes	There is a clear button at the bottom of the form. The user has to scroll to find this but a member of staff signing off a job will have to mark it as complete using fields at the bottom, so this should not be a concern.
Is it obvious how the control works?	Yes	Yes, it's a button and is easy to use.
Does the user understand the feedback they have been given?	Yes	Yes, the form quickly shows "Sync successfully scheduled" before the user is returned to the list of saved changes where the saved change has now been removed, given its successful upload to Appenate.

13. Sign out of the application

Table 80: Task A: Step 14 Test Retest Outcomes

Current Screen View: "Home"		
Frame	Heuristic Outcome	Outcome Comments
Does the user know what needs to be done?	Yes	They have completed the task they wanted and wish to sign out of the application. It is expected that the user understands they want to go back to "Home" to sign out of their account.
Can the user see a control to do that?	Yes	There is an arrow in the top left corner that looks as if it will be taking the user back to a previous step

Is it obvious how the control works?	Yes	Yes, it's a back icon so it is presumed that users will think that it is a button.
Does the user understand the feedback they have been given?	Yes	The user now finds themselves back at the "Home" screen where they are likely to now want to find a way to sign out.

10.10.2 Walkthrough B: Edit an Existing Component Order

1. Open the Application

This step has already been tested in a previous user task. See Table 68Table 77 for this step test.

2. Sign In using assigned Appenate credentials

This step has already been tested in a previous user task. See Table 69 for this step test.

3. Select the “Component Order” menu item

Table 81: Task B: Step 3 Test Outcomes

<i>Current Screen View: “Home”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, they want to view the status of a given order in the system by searching it up.
Can the user see a control to do that?	Yes	Yes, there is a button in the centre of the menu that shows a clock icon and the caption of “Component Orders”.
Is it obvious how the control works?	Yes	Yes, it’s a button control and should not be mistaken by the user. Very familiar.
Does the user understand the feedback they have been given?	Yes	A new page is shown on-screen. It should be assumed that the user understands the result of the button click action.

4. Select the Select the correct Component Order ID

Table 82: Task B: Step 4 Test Outcomes

<i>Current Screen View: “Component Order”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, the user wishes to select the precise order so that they can view its status and delivery details to be sure that it can be edited to provide the resources on site in time.

Can the user see a control to do that?	Yes	Yes, at the top of the screen there is a control showing “Edit or Create a Component Order”.
Is it obvious how the control works?	Yes	Underneath the caption it shows a blank field and the dropdown arrow icon. This suggests that options will appear below the control when pressed by the user.
Does the user understand the feedback they have been given?	Yes	A spinner wheel has appeared adjacent to the control that the user can ‘spin’ to select the desired order record. This is clearly an action of clicking the arrow.

5. Browse or update the various screen fields as desired involving scroll

This step has already been tested in a previous user task. See Table 68Table 77 for this step test.

6. Exit, not saving any changes made (none were made)

Table 83: Task B: Step 6 Test Outcomes

<i>Current Screen View: “Component Order”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	They want to go back so that they can find the contents of an order and amend them. The contents are not noted on this screen so they want to look elsewhere.
Can the user see a control to do that?	Yes	There is an “Exit” button in the top left hand of the screen. This looks most likely as the “Save” and “Create” do not match the user’s intentions.
Is it obvious how the control works?	Yes	Yes, it’s a button and suitably clear. It is located in a familiar screen location too.
Does the user understand the feedback they have been given?	Yes	Yes, a confirmation pop-up has appeared to check with the user that they want to exit without saving any changes. They do, as they have made no changes to this record.

7. Select the “Order Contents” menu item

Table 84: Task B: Step 7 Test Outcomes

<i>Current Screen View: “Home”</i>

<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes. Having checked the order status, they now want to edit its contents.
Can the user see a control to do that?	Yes	Yes, a button that reads "Order Contents" looks most likely on the home page.
Is it obvious how the control works?	Yes	Yes, it's clearly a button that a user can click
Does the user understand the feedback they have been given?	Yes	Yes, they have been taken to a new page entirely so this action is clear.

8. Select the correct order from the Order ID lookup

Table 85: Task B: Step 8 Test Outcomes

<i>Current Screen View: "Order Contents"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	They still wish to find the order and edit the component contents of it.
Can the user see a control to do that?	Yes	Yes, there is a text box at the top of the page which reads "Create or Edit Order Contents".
Is it obvious how the control works?	Yes	It should be assumed that a user will understand. The placeholder reads "Enter Search Keywords" and the text box will allow the user to search for an ID.
Does the user understand the feedback they have been given?	Yes	Yes, a keyboard has appeared prompting the user to enter an ID in this field. When typing an ID any number of orders, items placed under the desired ID appear. It is clear that this is a check box owing to the blue box at the right-hand side of the matched result.

9. Edit the existing component contained within using the Component ID field

Table 86: Task B: Step 9 Test Outcomes

<i>Current Screen View: “Order Contents”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	They have the selected record visible, but now want to edit the component contained within.
Can the user see a control to do that?	Yes	A user can see a Component ID field below the Order ID field.
Is it obvious how the control works?	Yes	It’s a text box and is showing text box content when an order is selected.
Does the user understand the feedback they have been given?	Yes	Yes, a keyboard is now being displayed having selected the field. A user can now remove the existing component and replace it with a new item identifier.

10. Save the changes made to the record and exit

Table 87: Task B: Step 10 Test Outcomes

<i>Current Screen View: “Order Contents”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	They now want to save the change that they have made to the existing record.
Can the user see a control to do that?	Yes	Yes, there is an “Exit” button in the top right corner of the screen.
Is it obvious how the control works?	Yes	Yes, this is clearly a button and is familiar to users.
Does the user understand the feedback they have been given?	Yes	Yes, a new pop-up box has appeared on screen. The result of attempting to exit has displayed options to save or discard changes made.

11. Click on “Order Contents” menu item

Table 88: Task B: Step 11 Test **Retest** Outcomes

<i>Current Screen View: “Home”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Limited by builder interface and erroneous upload functionality – unable to fix	The nature of the front-end builder considerably limits the changes that the developer can make to the processes of saving and uploading changes. Given the time taken for the developer to understand what they need to do it is expected that a user will struggle, at least when first using the application. If however a user does understand they haven’t yet completed the goal of uploading the record the following questions can be applied.
Can the user see a control to do that?	Yes	To access this menu again they can clearly see an “Order Contents” button.
Is it obvious how the control works?	Yes	Yes, it’s a button and is clear.
Does the user understand the feedback they have been given?	Yes	The screen view has changed and a new pop-up dialogue box is being shown to them.

12. Click “View Saved Entries”

Table 89: Task B: Step 12 Test Outcomes

<i>Current Screen View: “Order Contents”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Now they might become aware that they have not completed the overall goal and that the edited record must be uploaded or synced in some way.
Can the user see a control to do that?	Yes	Yes, there are two buttons on this pop-up menu. The user wishes to open a saved record so selects “View Saved”.
Is it obvious how the control works?	Yes	Yes, they are clearly buttons that the user must click.

Does the user understand the feedback they have been given?	Yes	Yes, a completely new screen is shown that contains a list of all changes made to table records.
--	-----	--

13. Select the correct edit instance from the returned list of edit instances

This step has already been tested in a previous user task. See Table 77 for this step test.

14. Click “Open” to view it

This step has already been tested in a previous user task. See Table 78 for this step test.

15. Click “Create” to upload the record

This step has already been tested in a previous user task. See Table 79 for this step test.

16. Go “Home” to sign out of the application.

This step has already been tested in a previous user task. See Table 80 for this step test.

10.10.3 Walkthrough C: Create a new Maintenance Job

1. Open the Application

This step has already been tested in a previous user task. See Table 68Table 77 for this step test.

2. Sign In using assigned Appenate credentials

This step has already been tested in a previous user task. See Table 69 for this step test.

3. Select the “Vehicle Maintenance” menu item

This step has already been tested in a previous user task. See Table 70 for this step test.

4. Select the correct Maintenance Job ID

This step has already been tested in a previous user task. See Table 71 for this step test.

5. Browse the various screen fields as desired involving scroll

Table 90: Task C: Step 5 Test Outcomes

<i>Current Screen View: “Vehicle Maintenance”</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Before entering a new job they are browsing the dates and work allocation of existing work. They know that they want to look at current work.
Can the user see a control to do that?	Yes, many	There are numerous fields that the user can scroll upwards and downwards to view. This should be clear to the user as several are off the page.
Is it obvious how the control works?	Yes	Yes, there is a scroll with which to view field contents and the spinner (already interacted with by the user) to change the record being viewed.
Does the user understand the feedback they have been given?	Yes	Yes, the page moves up and down and they can move between record views.

6. Empty the Job ID field to clear all fields

Table 91: Task C: Step 6 Test Outcomes

<i>Current Screen View: "Vehicle Maintenance"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	They want to empty all the fields to start inputting data to create a new work item. They know they don't want to edit an existing item.
Can the user see a control to do that?	Yes	Yes, the top control informs the user to "Select an ID or leave this field blank to create a new record".
Is it obvious how the control works?	Yes	Yes, there is a dropdown icon that suggests that the user must click for more options. Above the first ID a user can select " " – a blank ID for this field. Whilst this should be fairly easy for users to understand, the next step of client-based user testing will provide more insight into how users will react to this control.
Does the user understand the feedback they have been given?	Yes	All the fields are now blank, allowing the user to start entering new data into each field as they deem necessary.

7. Add in the desired fields so as to create the details for a new piece of work

Table 92: Task C: Step 7 Test Outcomes

<i>Current Screen View: "Vehicle Maintenance"</i>		
<i>Frame</i>	<i>Heuristic Outcome</i>	<i>Outcome Comments</i>
Does the user know what needs to be done?	Yes	Yes, they are now presented with empty fields that they can complete as appropriate for the new job.
Can the user see a control to do that?	Yes, many	There are numerous picker-style spinners that can be accessed by the conventional dropdown. Text boxes allowing keyboard-based text entry can also be used.
Is it obvious how the control works?	Yes	Yes, there are date picker spinners, vehicle and staff ID pickers and text boxes, all of which are common controls and have been identified above as being appropriate and suitably accessible.

Does the user understand the feedback they have been given?	Yes	Fields populate with new data having selected a dropdown menu item, selected a date or added content into a text-based field.
--	------------	---

8. Click “Save” to save updated record

This step has already been tested in a previous user task. See Table 74 for this step test.

9. Click on “Order Contents” menu item

This step has already been tested in a previous user task. See Table 84 for this step test.

10. Click “View Saved Entries”

This step has already been tested in a previous user task. See Table 89 for this step test.

11. Select the correct edit instance from the returned list of edit instances

This step has already been tested in a previous user task. See Table 77 for this step test.

12. Click “Open” to view it

This step has already been tested in a previous user task. See Table 78 for this step test.

13. Click “Create” to upload the record

This step has already been tested in a previous user task. See Table 79 for this step test.

14. Go “Home” to sign out of the application.

This step has already been tested in a previous user task. See Table 80 for this step test.

---End of Cognitive Walkthrough Scripts---