

UNIVERSITY OF WALES: TRINITY SAINT DAVID - SWANSEA
METROPOLITAN

Usability testing and Surveillance Applications using a Responsive Web Design approach

by

Jonathan Ieuan Evans

A project submitted in partial fulfillment for the
degree of B.Sc.(Hons) Software Engineering

in the
School of Applied Computing
Faculty of Architecture, Computing and Engineering

Supervisor: Michael Dacey

May 2019

Declaration of Authorship

I, [Jonathan Ieuan Evans](#) 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: Jonathan Evans

.....

Date: 1st May 2019

.....

Form of Consent

I [Jonathan Ieuan Evans](#) hereby consent that my Project, submitted in candidature for the B.Sc.(Hons) Software Engineering 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:Jonathan Evans

.....

Date: 1st May 2019

.....

UNIVERSITY OF WALES: TRINITY SAINT DAVID - SWANSEA
METROPOLITAN

Abstract

Faculty of Architecture, Computing and Engineering
School of Applied Computing

B.Sc.(Hons) Software Engineering

by Jonathan Ieuan Evans

Creating cross-platform interfaces can be a troublesome experience as multiple screen form factors must be catered for - such as Mobile and Desktop computers. It is desirable that a given interface would be considered usable on each platform, however, this may not be an easy task when using different windowing frameworks to create separate interfaces for use with devices.

This project takes Responsive Web Design (RWD) as an approach to producing a cross-platform surveillance application. The project was carried out in three phases, investigating relevant literature, producing a surveillance system interface, and testing the given interface in an iterative manner where different iterations will be created as a result of testing.

The outcome of this project is testing analysis which identifies potential usability problems with the created surveillance system interface, along with their resulting corrective actions. These created solutions may prove useful when creating future complicated interfaces (such as that of a surveillance system) using RWD.

Contents

Declaration of Authorship	i
Form of Consent	ii
Abstract	iii
1 Introduction	1
1.1 Aim	1
1.1.1 Objectives	1
1.2 Structure of the Report	2
2 Literature Review	3
2.1 Introduction	3
2.2 Usability Engineering and Testing Methodologies	4
2.2.1 A/B (Split) and Multivariate testing	4
2.2.2 Task-based testing (Task Scenarios)	5
2.2.3 Thinking aloud	5
2.2.4 Eye Tracking	5
2.2.5 Quantity of usability tests	6
2.2.6 Heuristic Evaluation	7
2.3 User Experience Design and HCI	7
2.3.1 User Experience Design and Accordances	7
2.3.2 Human Factors and Human Computer Interaction	8
2.3.3 Web Content Accessibility Guidelines	8
2.3.4 Shneiderman's Golden Rules of Interface Design	8
2.4 Desktop and Smart Devices	9
2.5 Responsive Web Design	10
2.5.1 Web Technologies - HTML	10
2.5.1.1 Semantics of HTML	11
2.5.2 Web Technologies - CSS	12
2.5.3 Web Technologies - JavaScript	12
2.6 Length Units and Layout methods in CSS	13
2.6.1 Length Units	13
2.6.1.1 Percentages, rem and em	13
2.6.1.2 Pixels	13
2.6.2 Layout methods	13
2.6.2.1 Float layouts	14
2.6.2.2 Flex-box	14

2.6.2.3	CSS Grids	14
2.6.2.4	Browser support	14
2.6.3	Media Queries	15
2.7	Existing Surveillance Applications	15
2.7.1	ISPYCONNECT	15
2.7.2	Nest Cameras	16
3	Research and Development Methodology	17
3.1	Research Methodology	17
3.2	Data Collection	17
3.2.1	Desktop Users Tests	18
3.2.2	Mobile Users Tests	18
3.3	Development Methodology	19
3.4	Development Tools	20
3.4.1	Software	20
3.4.2	Programming languages involved	21
3.4.3	Research Deliverables	21
4	Design and Implementation	22
4.1	Design	22
4.1.1	Low-Fidelity Wireframe	24
4.1.2	High-Fidelity Wireframe	25
4.2	Implementation	25
5	Analysis	31
5.1	Iteration #1	31
5.1.1	Iteration #1 - Desktop	31
5.2	Iteration #1 - Mobile	32
5.2.1	Iteration #1 - Corrective Actions	32
5.3	Iteration #2	32
5.3.1	Iteration #2 - Desktop	32
5.4	Iteration #2 - Mobile	32
5.4.1	Iteration #2 - Corrective Actions	32
5.5	Iteration #3	33
5.5.1	Iteration #3 - Desktop	33
5.5.2	Iteration #3 - Corrective Actions	33
6	Evaluation	34
7	Conclusions and Recommendations	36
7.1	Future Recommendation	36
8	Reflection	38
A	“Index” Page	39
B	“Live” Page	41
C	“Recordings” Page	43

D “FAQ” Page	48
E JavaScript	51
F Log book	63
 Bibliography	 73

Chapter 1

Introduction

This project uses Responsive Web Design (RWD) to create a cross-platform surveillance application accessible through devices with Web Browsers.

1.1 Aim

The aim of this project is to identify problems when creating cross-platform interfaces using RWD, and providing solutions to said problems.

1.1.1 Objectives

The aim of this project will be accomplished by meeting the following objectives:

- Carry out a literature review that investigates web technologies, usability engineering fundamentals, and existing surveillance applications.
- Create a first iteration of the surveillance application. This will provide the core components of the application – such as adding/removing cameras, browsing cameras.
- Produce a task list that will be provided to test subjects to carry out specific tasks on the application.

- Conduct a usability study utilising an eye tracker to discover insightful issues regarding usability issues on the application.
- Create more iterations as a result of usability studies. The process of testing an application and creating new iterations will be repeated until no or very few problems are discovered in the usability testing.

1.2 Structure of the Report

This report will consist of eight chapters. The contents of these chapters are explained in brief below.

- Introduction – Explains in brief, details about this project, including the aims and objectives.
- Research and Review of Literature – This examines literature pertaining to this project (Responsive Web Design, Usability Engineering, and Surveillance systems). It explains all the necessary knowledge needed to create the aforementioned surveillance system.
- Research and/or Development Methodology – The methodology taken to research and development are explained in detail in this section.
- Design and implementation – details how the surveillance system was implemented in regards to its design and development.
- Analysis – provides results from usability tests carried out as part of this project.
- Evaluation – Evaluates the objectives on the project
- Conclusions and Recommendations – concludes the report and adds recommendations for further research and development for potential further iterations.
- Reflections – Reflects on the process of creating this project

Chapter 2

Literature Review

2.1 Introduction

Computing technology is an ever-evolving industry. As of 2016, the majority of Internet usage has shifted from Desktop computers to Mobile devices [1]. There are currently “Smart Devices” that bring Operating System functionality to otherwise simple products, such as a mobile phone, or a watch, being superseded by devices such as Smart Phones and Smart Watches — most of which are capable of accessing the Internet through a Web Browser. With the introduction of these devices, a design problem has arisen as a result of such devices having Internet browsing capabilities, and that is, users are now using multiple devices for accessing their web-based services. As a result of this, layout techniques such as Responsive Web Design have been created which allow developers to create websites that can be scaled across multiple screen resolutions.

This literature review will investigate testing methodologies that will be used for testing an interface of a surveillance application across multiple platforms. A surveillance application is a suitable candidate for testing usability across these devices as such systems usually present a variety of features that may prove difficult when implementing them on smaller screens. After investigating testing methodologies, the properties which enable Responsive Web Design will be investigated. This will be concluded by brief research into existing surveillance systems.

2.2 Usability Engineering and Testing Methodologies

Usability Engineering is the field of computer science that corresponds to the study of usability of a software product. HCI Research and usability studies are commonly carried out to identify rules in which a system can be evaluated and designed. This section will investigate usability testing methods that can be used within user studies.

Usability testing consists of testing how a user interacts with a given application. There are a wide range of testing methodologies such as eye-tracking, A/B testing, focus groups, and surveys, all of which will provide qualitative (largely subjective) or quantitative (mostly objective and measurable) information that allow developers to make educated decisions on their design decisions when creating their interfaces [2].

2.2.1 A/B (Split) and Multivariate testing

A/B Testing (or Split testing) is a type of controlled experiment, commonly carried out on web sites. This experiment tests a portion of an audience of a product. [3] It is used for testing two versions of an interface to decide which achieves the goals desired by the creator of the interface. This may be done to improve conversion rates (the rate that a user becomes a customer by making purchases), or in-case of usability testing, finding out which interface is easier to use. The changes to the interface may be based on a hypothesis that a researcher may have — which may come from analytical data or just pondering.

In A/B testing, the two versions of an interface are given names, the control version, and the treatment version [4]. The control version is the version of the interface that may be used in production, and the treatment version is the version that has a changed variable from the control version.

Another experimentation method similar to split testing is known as Multivariate-Testing. This is when multiple variables are tested, contrary to the two that are used in A/B testing [5]. This may be desired by company's that can invest the resources into conducting such experiments as the process would be more timely as more than two variances are being tested.

Split testing is a proven method for improving conversion rates on-line (how many people brought a product). There are many case studies to read from, however, one relatively recent interesting case study conducted by Optimizely [6] shows how the Digital team for Barack Obama’s campaign website in 2012 improved conversion rates (or how many people donated) to the campaign by 49%, as well as email sign ups by 161%. This was done by conducting multiple A/B tests using the Optimizely platform and testing multiple hypotheses that the digital team had and carrying out the relevant changes where necessary.

2.2.2 Task-based testing (Task Scenarios)

Task-based testing is the process of getting a user to carry out a “Task Scenario” as described by Jakob Nielson [7]. This will provide insights into how users actually use a system. It consists of given a number of users scenarios to try out, and observing the outcomes.

2.2.3 Thinking aloud

“Thinking Aloud” is the term used to describe a test subject that talks about their immediate thoughts as they are thinking during a usability test. Neilson [8] suggests that thinking aloud can be beneficial when carried out alongside user tests when evaluating a user interface.

Ericsson et al. [9] detail the psychology behind “thinking aloud” to gain insights into the thought process of a test subject. The implication of this is that test subjects will be able to describe their thought process more clearly as they describe what they are doing at the time they are doing it. Probing for information from Short Term Memory in this way is an effective way to get unbiased information from a test subject.

2.2.4 Eye Tracking

Eye tracking research consists of using an eye tracking instrument to track the eyes of the user.

When an eye tracker is active, the test subject could be prompted to carry out task scenarios provided by the test conductor. This could be a list of tasks that must be

carried out. During this stage, the test conductor may wish to record some crucial information on paper, such as how many steps a given task takes.

Programs such as Ogama [10] can be used to retrieve data from test subjects such as gaze maps, which show where people are looking, and eye gaze playbacks. This software can be used in conjunction with eye-tracking devices such as the Tobii Eye tracker to retrieve this information.

Nielson advises the test subject can “Think Aloud” during these tests as described in the previous section, meaning that the user must speak their thought process as they carry out tasks. This is particularly insightful because it provides a glimpse into what the test subject is thinking and may uncover information that would not be uncovered otherwise.

2.2.5 Quantity of usability tests

Nielson [11] suggests that carrying out usability tests with just five users will uncover the majority of usability problems with a user interface. As the graph in **Figure 2.1** demonstrates, the chances of discovering new usability insights diminish greatly after the first five users are tested.

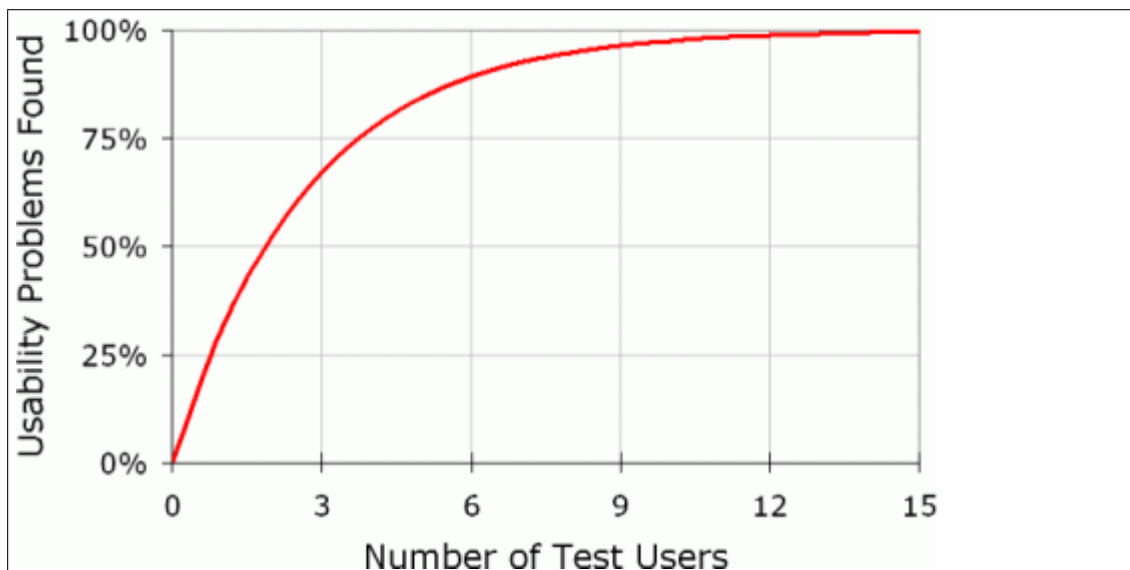


FIGURE 2.1: Diminishing returns illustrated by Nielson’s Graph [11]

2.2.6 Heuristic Evaluation

Heuristics as defined in Nielson and Molich's paper on Heuristic Evaluation [12] describes a set of principles on which a user grades (or evaluates) a user interface. The paper emphasizes that such an evaluation must be carried out with multiple subjects. These principles were later re-defined in the book Usability Engineering [13]. The re-defined principles are shown in the excerpt from the book in **Figure 2.2**.

- Speak the Users' language
- Minimize User Memory Load
- Consistency
- Feedback
- Clearly Marked Exits
- Shortcuts
- Good Error Messages
- Prevent Errors
- Help and Documentation
- Simple and Natural Dialogue

FIGURE 2.2: Excerpt on Heuristics from Usability Engineering

2.3 User Experience Design and HCI

User Experience (UX) Design is concerned with creating an experience for users. The need for design in surveillance systems is required as most systems are currently targeting Desktop User Interfaces (UI). Because of this, only Desktop users can effectively use such systems, which reduces the potential user base substantially as Desktop users no longer account for the majority market share in computing — which have been claimed by mobile computers [14].

2.3.1 User Experience Design and Accordances

User Experience is a term coined by Don Norman, who is the first person to have the job title UX designer. He is responsible for a large amount of terminology in UX design.

Affordances and Signifiers are key elements of usability introduced by Don Norman [15].

An affordance describes the relationship between how a user interacts with an object, such as touching, pushing, and pulling. Norman describes this relationship such that, for example, “a door could afford pushing”.

Signifiers complement affordances in that they signify where an affordance can take place. A popular example of this, as described in Norman’s book, a door could have a rectangle plate that signifies that the door can be pushed. As a result, doors that implement a rectangle plate on one side, and a bar to pull on the other leave very little room for ambiguity, thus the user will be able to work out quickly how the door is used.

2.3.2 Human Factors and Human Computer Interaction

Human Factors is concerned with how physical attributes of a person can be applied in the design process of a given application.

Human Computer Interaction is a field of Computer Science that studies the psychology of how a user interacts with computer interfaces. Information from HCI research can be applied in the design process of a website’s creation to ensure that educated decisions are made.

2.3.3 Web Content Accessibility Guidelines

The WCAG (Web Content Accessibility Guidelines) is a set of accessibility rules detailed by the W3C (World Wide Web Consortium). Following these guidelines will produce a design that is considered accessible by many people — including those who have difficulty with eyesight, colour-blindness, and deafness [16].

2.3.4 Shneiderman’s Golden Rules of Interface Design

When designing a user interface, it may be useful to follow a set of principles that strive for great usability. Shneiderman [17], a notable professional in the field of human computer interaction, outlines eight rules for interface design as shown in **Figure 2.3**

- Strive for consistency
- Seek universal usability
- Offer informative feedback
- Design dialogs to yield closure
- Prevent errors
- Permit easy reversal of actions
- Keep users in control
- Reduce short-term memory load

FIGURE 2.3: Eight Golden Rules of Interface Design from Designing The User Interface

As described by Shneiderman, these rules are used by students and designers for creating interfaces and are generally well accepted amongst HCI professionals.

2.4 Desktop and Smart Devices

Usage of mobile devices have over-taken Desktop usage in 2016 as shown by usage statistics collected by StatCounter [1]. Considering this, development of applications for smart devices can not be ignored as a large market exists for a company to target. It may be appealing to write applications for these devices, however this increases the amount of programming required to support these devices in addition to desktop computers, of which include different operating systems with their own windowing solutions such as Android, iOS, Windows, Mac, and UNIX based-systems.

In order to support these devices a developer can alternatively opt to support applications that run within the Web Browsers of these devices. There are many devices capable of accessing the Internet which are loaded with web browsers at present. As a result of this, developers can write Web Applications where they will get to benefit from only supporting one main platform, web browsers. Even although there are multiple browsers, the language for writing interfaces is still the same — HTML (Hyper Text Mark-up Language), CSS (Cascading Style Sheets) — which can be used to create responsive web layouts (which is discussed in the following section), and JavaScript.

2.5 Responsive Web Design

Responsive Web Design is a website development approach for creating website layouts using CSS (Cascading Style Sheet — a language for defining website styles, that is capable of scaling to different resolutions).

Having a website that is capable of re-scaling when needed is ideal compared to the alternative option, which is to design for specific resolutions as it is possible that a device manufacturer will create a device which has a resolution that does not work well with a given interface — this is why a need for responsive web pages are needed.

The main technologies for creating a website are HTML, CSS, and JavaScript. These are used for front-end development, where the development carried out is concerned with how the website looks, contrary to back-end development, where the development is concerned with the process in how pages are served.

2.5.1 Web Technologies - HTML

HTML (Hyper Text Mark-up Language) is the language used to define elements on a page. It is very similar to XML (Extensible Mark-up Language) in that it follows the same format for describing it's elements.

```
<!DOCTYPE html>
<html>
<head>
<title>My website</title>
</head>
<body>
<!-- a comment -->
</body>
</html>
```

LISTING 2.1: Snippet of HTML

2.5.1.1 Semantics of HTML

It is worth noting that the semantics of HTML is crucial to describing how a page looks. HTML elements are described using tags. HTML5 in recent years has become a recommended specification by W3 [18]. The HTML specifications describe how semantics in the language should work. In particular HTML5 introduces new elements that allow developers to describe common elements on websites - such as *nav*, for navigation, *section*, for sections, and *article* for articles.

With improved semantics, programs such as screen readers and web crawlers (a program that traverses web pages, identifying the site map and relevant links for indexing on a search engine), can more efficiently represent sections of a website. **Figure 2.4** illustrates how a text-only browser would display HTML. This is useful as it gives a clear indicator how a web crawler may see it.

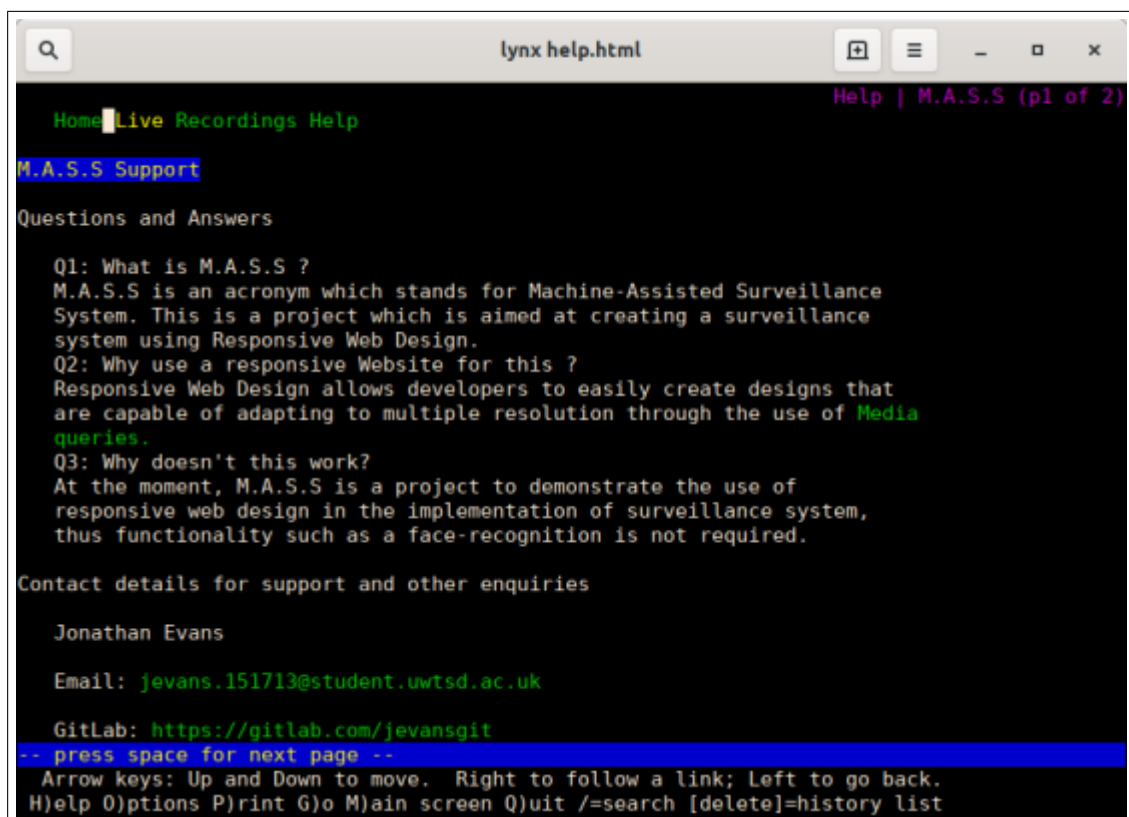


FIGURE 2.4: Lynx Command Line Interface Browser demonstrating a hierarchical view of a HTML document

2.5.2 Web Technologies - CSS

CSS (Cascading Style Sheets) allow developers to specify how elements of a page in HTML should be styled given a set of properties, such as, colours, sizes, and visibility properties. In particular, with CSS, a developer can create layouts that allow for unique layouts using properties to portion sections of a website.

2.5.3 Web Technologies - JavaScript

JavaScript is a scripting language that allows developers to embed programming code within their websites which is capable of manipulating the DOM (Document Object Model, the hierarchy of objects displayed in HTML). This can be used in conjunction with third-party libraries such as React.JS (React) for creating interfaces where the objects displayed on page must be manipulated.

React in particular is useful for creating interfaces that “react” to changes within data on a page, re-displaying the page with the updated data. It also utilises a concept known as a “Virtual DOM”, which is effectively an in-memory “copy” of the DOM. This provides efficiency as the JavaScript can be used to calculate the difference between the Virtual DOM, and actual DOM. As a result, React will know the most effective way to update the page with the least possible operations. [19]

2.6 Length Units and Layout methods in CSS

There are multiple methods used for creating layouts in CSS. Three common methods include floating elements, flex-box, and CSS grids [20] which are discussed in the following sub-sections.

In regards to Responsive Web Design, it should be noted that a website by itself is responsive when no modifications are made as no pixel measurements are specified. With layouts, it is common practice to define sizes of blocks on a website using units of length, of which, units such as percentages, em, rem, and pixels are used.

2.6.1 Length Units

Length units are used to specify lengths of elements on a page. These are crucial to acknowledge when implementing a responsive layout.

2.6.1.1 Percentages, rem and em

A website with no style will occupy 100% of screen width. Knowing this, websites can also be used to fit their width to a given percentage. The em and rem units can be used in addition to percentage to specify a size of an element that is relative to the root element.

2.6.1.2 Pixels

Contrary to percentages, pixels can be used for specifying a fixed (or static) layout.

When a layout relies solely on pixels, the website may only be usable at the resolution the website was created for. As a result, if the resolution of the device browsing the website is lower than the resolution the website was designed for, horizontal and vertical scroll bars will be present - which can be troublesome on mobile devices as users are required to pan the view port of the device using their fingers. One of the following sections describes how media queries in CSS can be used to remedy this.

2.6.2 Layout methods

Developers have created multiple methods for creating interfaces for websites. In the inception of web site development, it was not uncommon to use tables to create layouts,

however, advances have allowed developers to create layouts that are easier to create and can be described in a semantic way. Of which, three common methods are detailed below.

2.6.2.1 Float layouts

Float-based layouts use the CSS property *float* to “float” (or align) elements the right or left of the page. This can be used in combination with the *width* and *clear* property to specify the width of the property and if the alignment of elements should be broken or not (which will force an element to display on a new line, rather than in-line).

2.6.2.2 Flex-box

Flex-box is a relatively modern layout method used to describe layouts on a horizontal axis. [21] This allows developers to create horizontal grids in terms of flex-box units, rather than pixels.

2.6.2.3 CSS Grids

CSS Grids, similar to flex-box allows developers to create easy layouts on the horizontal axis, in addition to the vertical axis. [22] This framework allows developers to dynamically change the layout of the website by changing their position within a CSS grid.

2.6.2.4 Browser support

As more properties are added to the CSS specification, browser support becomes an issue as users will be required to have the latest browsers to use the latest CSS rules.

Acknowledging this, Flex-box may be desirable over the use of CSS Grids currently - solely due to CSS grids being a relatively recent development which currently has less than 93% browser coverage (92.33% [23]) on the Internet, where-as, flex-box supports 98.58% [24] of browsers on the Internet.

2.6.3 Media Queries

Media queries were introduced into the W3C CSS specification as a recommended standard in 2012 [25]. This function allows developers to change styles defined at different resolutions. The code snippet below demonstrates how a developer may wish to implement such behaviour.

```
@media(max-width: 960px)
{
    // Resize elements
    nav { width: 100%; }
    ...
}
```

LISTING 2.2: Media Queries in CSS

This is what enables the concept of a “Responsive website”, which is, a website that can be scaled to and from any resolutions, maintaining a usable interface through the use of media queries.

2.7 Existing Surveillance Applications

Most network camera software either alleviates windowing technology such as QT or Windows Forms, or in case of web-based software, SWF (Shockwave Flash) files or a HTML5 website. This section will investigate two popular surveillance applications and will identify their core components.

2.7.1 ISPYCONNECT

ISPYCONNECT [26] is an open-source surveillance application that makes use of Responsive Web Design and HTML5 elements, thus it is accessible on modern browsers. This software comes closest to what this project wants to implement as it is already a surveillance system implemented using Responsive Web Design.

- Timeline view
- Monitoring of networked cameras
- Management of recorded video files

FIGURE 2.5: Main features of ISPYCONNECT

2.7.2 Nest Cameras

Nest is a company that specialises in home automation. A popular product that they create are Nest Cameras. [27] These allow for users to easily install said cameras within a home for monitoring. The user interacts with the cameras through a mobile application. This application has the following capabilities:

- 24/7 Live Video
- Mobile phone alerts
- “Snapshot” history which describes what have may have happened at a given time
- “Person Alerts” which is used to communicate with people who are close to the cameras
- Recognises friendly people who are non-threats

FIGURE 2.6: Main features of Nest Cameras

Meanwhile Nest Cameras provide plentiful features and are suitable for any person to set-up on their premises, it still relies on a Windowing Framework (which is not detailed on-line). Meanwhile this provides great performance for the users of this product, it can prove challenging for developers when producing alternative interfaces (as they need to target more windowing frameworks).

Chapter 3

Research and Development Methodology

This project consists of data collection and software development. Data collection is used to collect data from usability tests that have been conducted, meanwhile, software development is used to create the product in which the usability tests are carried out on.

3.1 Research Methodology

The purpose of this project is to analyse and identify insightful information pertaining to how a cross-platform interface can be implemented. This takes a phenomenological approach – as the resulting data will require data collection, analysing, and implementation of changes that have been identified as a result of analysing data.

3.2 Data Collection

The data collection component of this project will consist of retrieving data obtained from a task list that will be given to a test subject, providing them with tasks to complete.

After a test subject is given a task list, they will be prompted to begin carrying out their tasks. When this had begun, Ogama (Eye-tracking software), started recording data. The data provided by the eye tracking software is quantitative data which describes where the user looks at a given time, and the directions where they are moving their mouse in.

The task list provided to the test subjects is shown below. This contains a list of task for users of mobile devices, and users of desktop devices. As Ogama does not currently support mobile devices, the browser window had to be re-sized to simulate the "look" of a mobile device.

3.2.1 Desktop Users Tests

Task No	Task Description
1	Add a camera
2	Name a camera
3	Add 5 more cameras
4	Navigate through each camera
5	Remove a camera
6	Browse recordings
7	Download a recording
8	Remove a recording
9	Edit information on a recording
10	Close camera adding window

3.2.2 Mobile Users Tests

Task No	Task Description
1	Add a camera
2	Name a camera
3	Navigate cameras
4	Remove a camera
5	Browse recordings
6	Download a recording
7	Remove a recording
9	Close camera adding window
8	Edit information on a recording
10	Open mobile menu
11	Close mobile menu

In addition to the quantitative data provided by Ogama, qualitative data was obtained while the tests were being carried out, by prompting the user to describe what their immediate thought process was as they were carrying out tasks.

After data had been collected, it would then have been analysed and any additional improvements would have been identified for the proceeding iteration.

3.3 Development Methodology

The development of this project will take the agile development approach.

Agile software development consists of incorporating development methods that result in a development process that allows developers to develop an application in a fast and iterative manner.

A looping spiral illustration is commonly used to describe agile development, which ultimately consists of development, testing, and producing a new iteration. An example of this process is shown in **Figure 3.1**.

This method has been taken as the project requires data to be gathered early-on. As a result, this provides the developer with a benefit, which is, changes can be made early in the product's development. This has been chosen as it prevents the need for making changes when the code base has grown to a point where the code may become unmanageable, thus making changes harder to make.

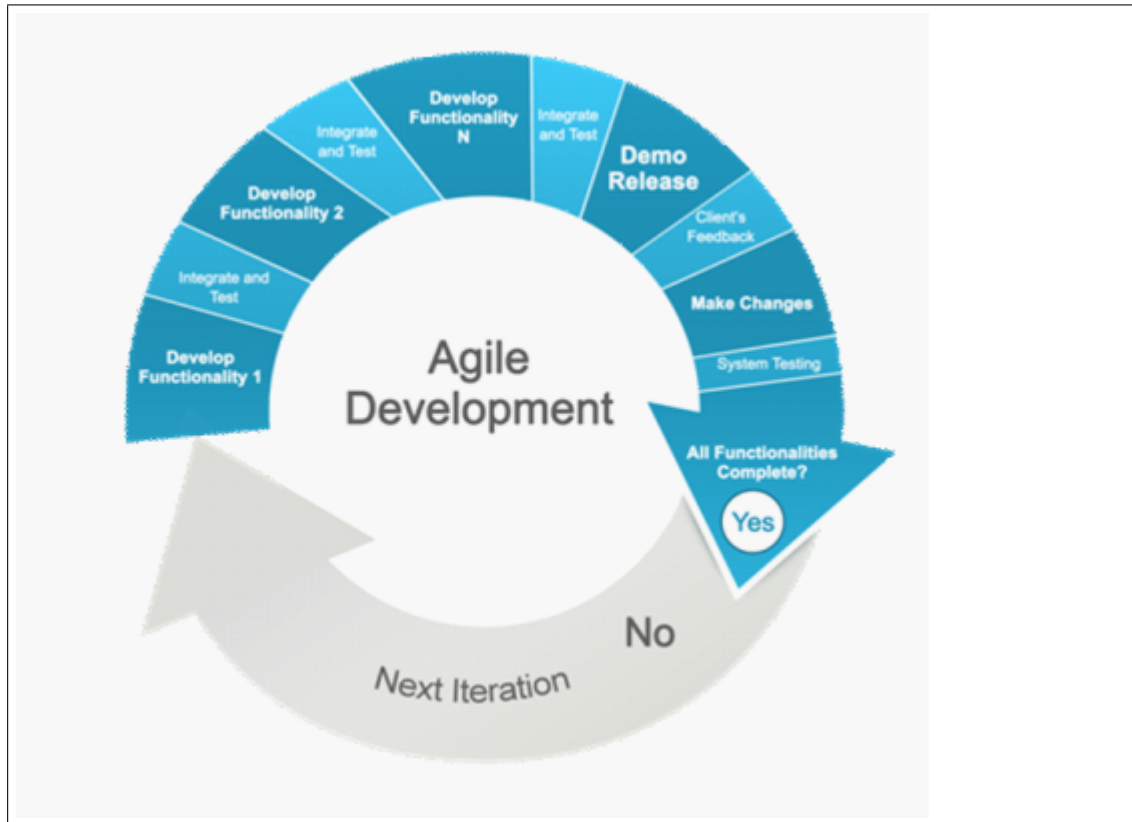


FIGURE 3.1: Agile Development Spiral (taken from <http://codigodelsur.com>)

3.4 Development Tools

The creation of this program utilised a modern Desktop computer with the GNU/Linux Operating System (however, packages for creation of the project are also available on Mac and Windows).

3.4.1 Software

- GitLab – This is a VCS (Version Control System). A VCS allows developers to create a repository in which their source code is stored, in addition to this, snapshots are made when a developer “commits” a change. As a result, all committed changes within the project can be “skipped to” to show what the source-code looked at a given time. Git is an example of a popular VCS solution. GitLab is a platform that allows developers to use Git for their projects. This was used as it offered free private repositories at the time of creation.

- Visual Studio Code – This is a text editor with IDE (Integrated Development Environment) features. Some note-worthy features that it includes are: Intellisense (auto-completing code / code suggestions), syntax highlighting, and code validation through use of plugins.
- SASS (Syntactically Awesome Style Sheets) – This is a CSS pre-processor. A pre-processor such as this allows developers to utilise functionality that is not present in standard CSS, such as variables, easier code nesting, and “mixins” – which provide a way to easily create re-usable code through inheritance.
- Gulp - this is “a streaming build system” that allows developers to carry out build tasks when the “gulp” program is called. This was used to compile SASS files. The reason for using this is that additional tasks could also be added a later date if desired.

3.4.2 Programming languages involved

The following languages were used

- HTML (Hyper Text Mark-up language) – Used for creation of HTML pages.
- CSS (Cascading Style Sheets) – for styling components of web pages.
- SASS – for enabling CSS pre-processing functionalities (as described above).
- JS (JavaScript) – this was used for simulating processing in the web application. Node packages were also used to retrieve software that enabled task automation (through use of Gulp as described above).

3.4.3 Research Deliverables

This project delivers two deliverables – an interface for a responsive surveillance system, and usability test analysis that will illustrate what changes were necessary to guarantee that the user interface was in-fact “usable”.

Chapter 4

Design and Implementation

This chapter covers details pertaining to the design and development of the finished product.

4.1 Design

Considering what was discovered about surveillance systems within the literature review, the design of the resulting surveillance system started with defining the requirements for a surveillance system.

The identified requirements are as followed.

The system should allow the user to:

- View cameras
- Read and delete information pertaining to the cameras (such as being able to access camera feeds, and remove them).
- Create, Read, Update, and Destroy information pertaining to incidents that have been recorded by cameras.

- Find information to help with any potential issues (a frequently asked question page).
- Contact information to get support if desired.

Knowing what was required of the website, low-fidelity wireframes were created, which were preceded by high-fidelity wireframes. At this point, the design details could be altered easily as no code has been written and the design only existed on paper. These wireframes are shown on the following page.

4.1.1 Low-Fidelity Wireframe

Figure 4.1 shows a low fidelity wireframe. Low-fidelity implies that any change made to this design will not require much work. At this stage, decisions have already been made about the placement of certain components within this application, such as where the menu is going, camera feed, and recordings.

This design would make use of three well recognised design patterns, a global navigation (a location where navigation will always be present, thus the user is able to learn where to find the navigation relatively quickly), a hamburger icon (a well-recognised image on mobile devices which users press to access a menu). The third pattern is known as diagonal balance – this is the weightings used on the page to emphasise different components of it's structure.

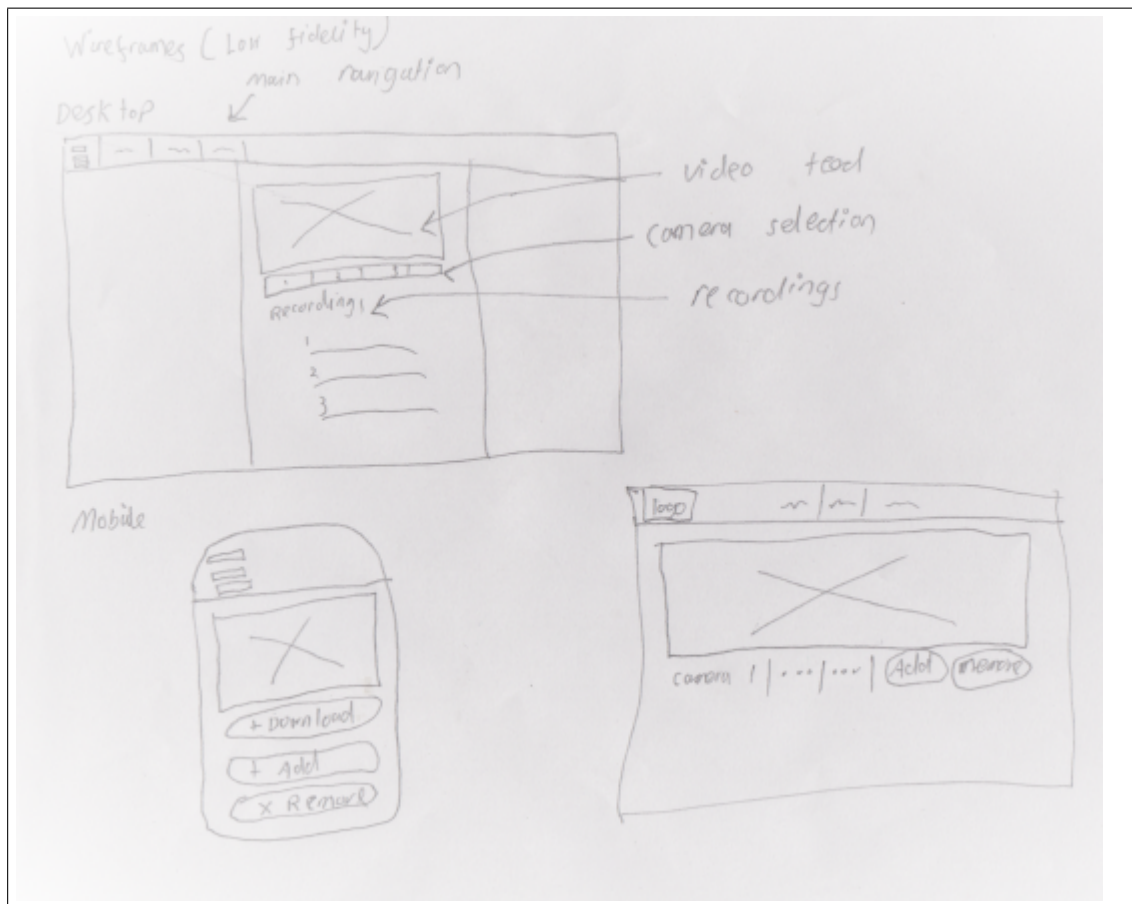


FIGURE 4.1: Low-Fidelity Wireframe

4.1.2 High-Fidelity Wireframe

This is the corresponding high-fidelity wireframe. It provides a detailed image of how the interface is supposed to look. The image itself shows the interface “in-action” where the user is scanning for available cameras on their network.

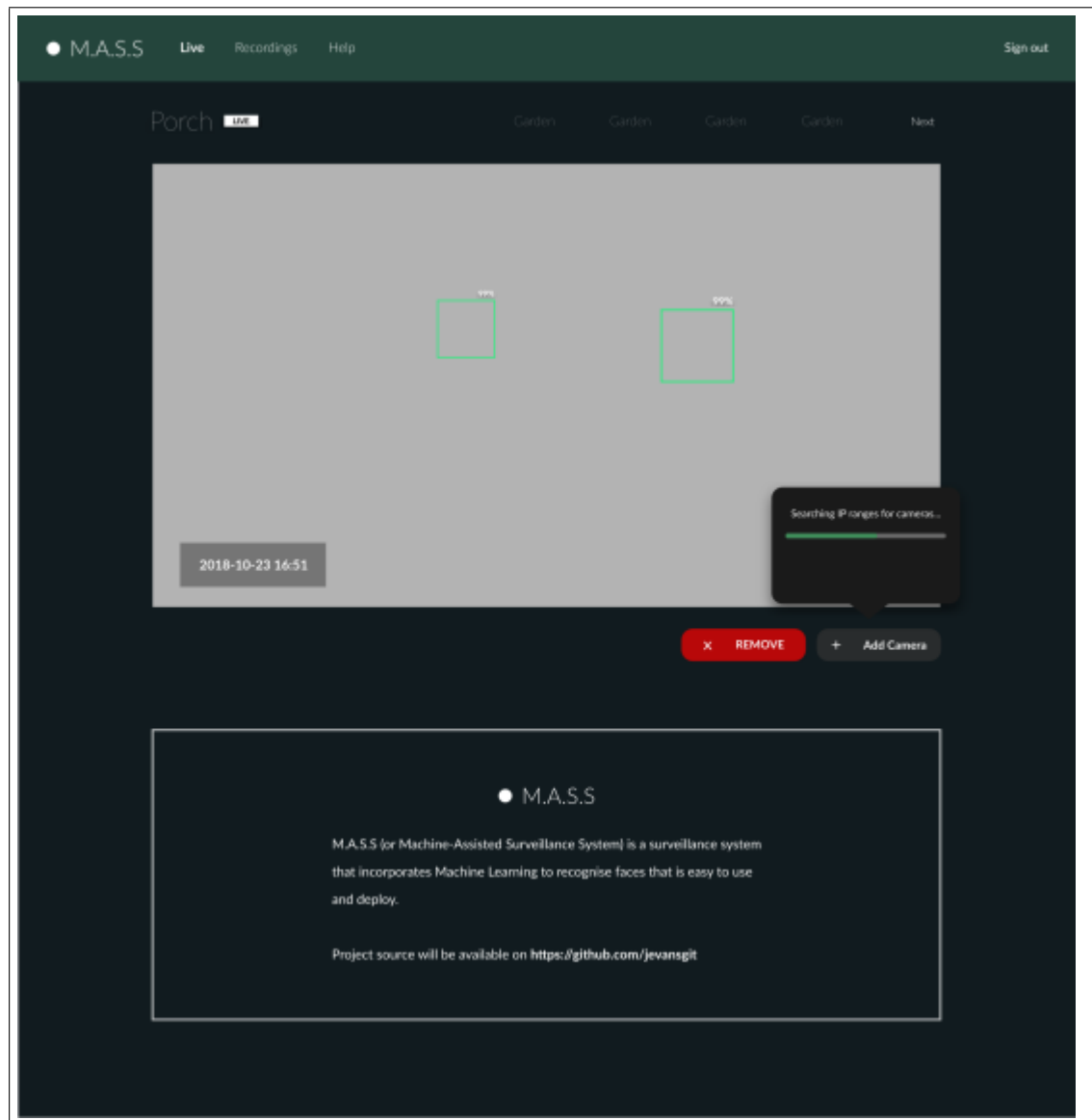


FIGURE 4.2: High-Fidelity Wireframe

4.2 Implementation

After the design phase, the final product was built in iterations, where-as, the first iteration would provide the base functionalities as described in the previous section, and

the following iterations were based on results obtained from usability studies carried out on different iterations of the product.

The website itself is a static page that utilises core web technologies such as HTML, CSS, JavaScript.

The JavaScript in particular was used to provide “dummy data” to the page to make it seem as if the page was retrieving data from IP (networked) cameras, where-as instead of this, the program would be faking this interaction as a true implementation of this was not needed to just test the usability of this interface.

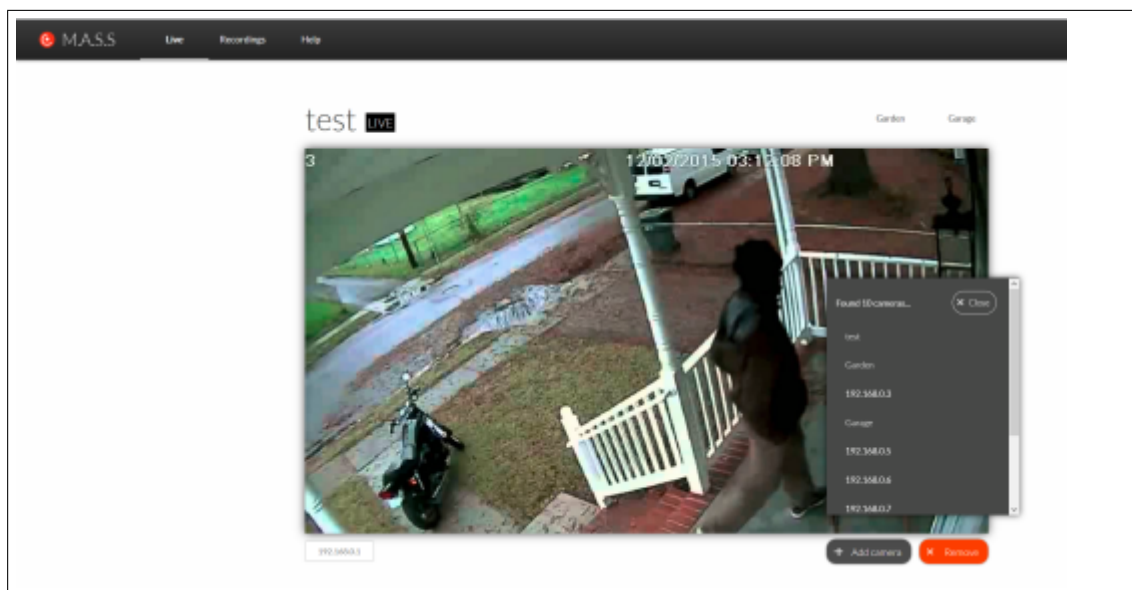


FIGURE 4.3: Picture of finished website

The following shows a snippet of JavaScript that shows briefly what the “Add cameras” button does in the application. Much of the other buttons work in a similar fashion; however, it became apparent after the code base grew, that an alternative structure was needed for adding more buttons and reducing code re-use. This has not been added in any current iterations however, but an implementation of this could be useful for later iterations.

```
if (btnAdd != null) {  
  btnAdd.onclick = function () {  
  
    if (document.querySelector('.camera-img') == null) {
```

```
windowAdd.style.marginTop = "30px";
windowAdd.style.marginLeft = "4px";
} else {
windowAdd.style.marginLeft = "0px";
}

let progress = 0;

if (windowAdd.classList.contains("on")) {
windowAdd.classList.remove("on");
console.log("* Turning off camera discovery ... *")
} else {
windowAdd.classList.add("on");
console.log("* Opening camera discovery ... *")
var i = 1;

if (!scanAlreadyRun) {
(function sleep(i) {
setTimeout(function () {
scanAlreadyRun = true;
progress++;
infoBtnAdd.textContent = "Searching IP ranges for cameras...
                        " + progress + "%";
progressBtnAdd.style.width = progress + "%";
if (--i) sleep(i)
else {
progressBtnAdd.style.background = "transparent";
progressBar.style.background = "transparent";
if (document.querySelector('.camera-img') == null) {
windowAdd.style.marginTop = "30px";
} else { windowAdd.style.marginTop = "-380px"; }
windowAdd.style.height = "350px";
windowAdd.style.overflowY = "auto";

for (let i = 0; i < 10; i++) {
var cameraItem = document.createElement("div");

var camera = document.createElement("div");
var cameraName = document.createTextNode("192.168.0." + (i + 1));
camera.appendChild(cameraName);
camera.className = "camera";

cameraItem.appendChild(camera);
cameraItem.className = "camera-item";
windowAdd.append(cameraItem);
}

var cameras = document.querySelectorAll('.camera-item');
// Add event listeners to each camera item
```

```
for (let i = 0; i < cameras.length; i++) {
  cameras[i].addEventListener('click', function(event) {
    console.log(i);
    var txtAdd = document.createElement("input");
    txtAdd.setAttribute("type", "text");
    txtAdd.className = "txtCamera";
    txtAdd.setAttribute("placeholder", "Enter camera name here...");

    var btnSave = document.createElement("div");
    btnSave.className = "btnSave"
    btnSaveName = document.createTextNode("Add");
    btnSave.append(btnSaveName);

    txtAdd.addEventListener('keyup', function(event) {
      if (event.keyCode === 13) {
        btnSave.click();
      }
    })

    btnSave.addEventListener('click', function(event) {
      console.log("Saving " + txtAdd.value);
      cameras[i].textContent = txtAdd.value;
      cameras[i].style.color = "#ccc";
      cameras[i].style.cursor = "default";

      var notification = document.createElement("div");
      var notificationText = document.createTextNode(txtAdd.value +
        " camera has been added");
      notification.appendChild(notificationText);
      notification.className = "notification notification--add";
      document.body.appendChild(notification);
      var currentImage = "img/streams/" + (i + 1) + ".jpg";

      cameraStream.push ({
        ip: "192.168.0." + (i + 1),
        name: txtAdd.value,
        imageSrc: currentImage
      });

      console.log(cameraStream);

      txtAdd.remove();
      btnSave.remove();

      let existingImage = document.querySelector('.camera-img');
      var cameraVideo = document.querySelector('.camera-video');
```

```
if (existingImage === null) {
  let camActions = document.querySelector('.camera-actions');
  camActions.style.padding = "0";
  camActions.style.border = "none"
  camActions.style.textAlign = "left";

  windowAdd.style.marginTop = "-450px";
  console.log("Image not found, adding");

}
refreshCameras();
});

// Use insertBefore to add textbox after a given camera item
let parentDiv = cameras[i].parentNode;
parentDiv.insertBefore(txtAdd, cameras[i + 1]);
parentDiv.insertBefore(btnSave, cameras[i + 1]);

});
}

blockingMenu = true;

infoBtnAdd.textContent = "Found 10 cameras...";
var btnClose = document.createElement("a");
var btnCloseText = document.createTextNode("Close");
btnClose.appendChild(btnCloseText);
btnClose.className = "button button--close";
btnClose.style.display = "block";
windowAdd.append(btnClose);

btnClose.onclick = function () {
  windowAdd.classList.remove("on");
}
};
}, 10);
})(100);
}
}

windowAdd.style.marginLeft = "0px";
}
}
```

LISTING 4.1: Snippet of HTML

For full code listings, see Appendices A-E.

Chapter 5

Analysis

During each iteration, analysis was carried out to identify further changes that can be made to the interface.

Below is the analysis of the data that was produced by Ogama eye tracking software and utilising the “think aloud” method as the test was being carried out.

Please note: Raw eye-tracking data is available on-request. View front page for contact information.

5.1 Iteration #1

Test subject: Robbie

5.1.1 Iteration #1 - Desktop

When carrying out this test, a few usability issues were discovered. Namely, the CCTV camera location was not disclosed on the recordings page. This may make it difficult for the user to understand what location they are looking at.

In addition to this, the location in which the user downloads the recorded clips did not stand out against the background colour, thus making it difficult to identify a pattern on the page where the “actions” pertaining to a given clip is located.

5.2 Iteration #1 - Mobile

The mobile tests were successful. The location of the “hamburger icon” was a personal hypothesis that was tested.

For this application, locating the “hamburger” on the left of the mobile navigation menu proved to be usable, as opposed to positioning it on the right. However, a true test for this would require eye tracking of someone physically holding a mobile device, rather than the simulated mobile interface that was provided by a Desktop web browser.

5.2.1 Iteration #1 - Corrective Actions

This test warranted two important changes - Adding the names to the locations in which recordings were made, and adding emphasis to the location in which the user can download recorded clips.

5.3 Iteration #2

Test subject: Jonathan

5.3.1 Iteration #2 - Desktop

This test identified that the “Download” button on the recordings page did not stand out against the background. This is relating to the actual presentation of the button itself, rather than the box the recording’s actions (including the button) were located on. In short, the button did not look like a physical button.

5.4 Iteration #2 - Mobile

No further usability issues identified

5.4.1 Iteration #2 - Corrective Actions

A potential solution to make a “Download” button look more like a physical button would be to apply a gradient and perhaps a shadow to make it look as if it is a component that can be clicked (or pressed in-case of a real-world button). This change is exactly what was implemented for the third iteration.

5.5 Iteration #3

Test subject: Ieuan

5.5.1 Iteration #3 - Desktop

This test had few usability issues, a note-worthy problem that occurred was that it was not apparent when the camera was added.

5.5.2 Iteration #3 - Corrective Actions

A corrective action that could be applied is to include some animation (a visual cue) to make newly added cameras easily identifiable. An animation or other visual cue has not been identified yet as said visual cue has not been decided on.

Chapter 6

Evaluation

This chapter will evaluate the project against a set of goals.

The main aim of this project was to produce a web-based surveillance system that is accessible across multiple platforms. The key feature of this project is that such a system should be usable with the design issues that arise when creating cross-platform interfaces.

The data gathered from the usability tests uncovered a few usability problems that were addressed in later iterations of the website. More errors could have in-cured if the usability features used (such as a clear interface hierarchy, global navigation, and well-recognizable icons such as a hamburger or plus icon) were not familiar to the test subjects. Individuals have become used to said conventions when browsing the Internet through their Desktop and Mobile devices.

This project has turned out to be a success as it has met the main objective: the creation of a user-friendly cross-platform surveillance system. The final website has very few usability issues as many have already been identified and addressed through use of usability tests as shown in the previous section.

If this project was to be carried out again, a few changes are advisable. The main two changes that could be made is establishing dates in which users can carry out user

testing (to allow them to make time for this task), and to use a third-party JavaScript library such as React JS to handle common JavaScript tasks such as updating the DOM (Document Object Model) efficiently. In addition, “linting” (Program analysis software) can be used to provide the developer with useful code suggestions.

Chapter 7

Conclusions and Recommendations

Concluding this report, objectives were met and a responsive surveillance system interface was created. This system utilises well-known usability features, such as a global navigation menu, hamburger icon for mobile navigation, and fitts' law (where adequate padding has been used to ensure user's do not mis-click an item). This resulting usability test show that the interface has relatively low usability issues, thus proving that the system provided demonstrates a capable interface that can be scaled across multiple platforms (Desktop and Mobile).

This project utilises two research methods, "thinking aloud" while carrying out tasks using an eye tracker. Meanwhile this provides adequate results for usability testing, more research methods can be used - such as A/B split testing as discussed in the Literature Review. This method was not used as it is only effective where a large user base can test the system.

7.1 Future Recommendation

As the provided product of this report is a website, it may be possible to utilise a technology known as "electron" - this provides a lightweight browser for web apps. This may be slightly quicker than utilising a full browser, however, it may bring marginal

improvements and will need to be tested as the performance improvements may not be worth extra work.

The final recommendation would be to find test subjects whose internet browsing capabilities are different from the main demographic, this may incorporate users that are colour blind, blind, or who may have trouble reading. Meanwhile the current product is tested to work with the main demographic, this group of people cannot be ignored as it accounts for a lot of Internet users.

Chapter 8

Reflection

When reflecting on the project, I am happy with the final product that has been created as it meets the goals outlined at the beginning of this report, however, there are a few things that I would do differently if I was to carry out a project similar to this in future.

Time management is a skill that I need to enforce more rigorously. Meanwhile a project plan had been created for this project, I had not been following it as closely as I should have - this resulted in some work that had been rushed. In future, I will use a time management application when working on a long-term goal, such as working on another project.

In addition to improved time management, I would also study JavaScript in more detail. This had been my first time writing JavaScript, thus knowledge of the language was quite new, as a result, I believe the final code could be made a lot more manageable and readable through utilising a JavaScript library such as ReactJS, as it provides functionalities for building feature-rich interfaces relatively quickly, as described in the Research and Development section of this report.

Appendix A

“Index” Page

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="description" content="Easy Machine-Assisted Surveillance">
  <meta name="theme-color" content="#3c3c3c" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" href="favicon.png">
  <link rel="stylesheet" href="css/main.css">
  <link rel="icon" href="favicon.png">
  <link href="https://fonts.googleapis.com/css?family=Lato:300,400" rel="
    stylesheet">
  <title>M.A.S.S</title>
</head>

<body id="index">
  <div class="mobile-nav">
    <div class="hamburger">
      <div class="bar"></div>
      <div class="bar"></div>
      <div class="bar"></div>
    </div>
  </div>
  <nav class="nav">
    <a class="nav__link nav__link__active" href="index.html">Home</a>
    <a class="nav__link" href="live.html">Live</a>
    <a class="nav__link" href="recordings.html">Recordings</a>
    <a class="nav__link" href="help.html">Help</a>
  </nav>
```

```
<div id="page">
  <div class="login-box">
    <h1>M.A.S.S</h1>
    <form method="get" action="live.html">
      <input type="text" placeholder="Please enter your username here">
      <input type="password" placeholder="Enter your password here">
      <input type="submit" value="Login" class="button button--login">
    </form>
    <p>M.A.S.S (or Machine-Assisted Surveillance System) is a Surveillance
system that incorporates Machine Learning to
      recognise faces that is easy to use and deploy.</p>
    </div>
  </div>
  <script src="js/main.js"></script>
</body>

</html>
```

“Index” Page

Appendix B

“Live” Page

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="description" content="Easy Machine-Assisted Surveillance">
  <meta name="theme-color" content="#3c3c3c" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" href="favicon.png">
  <link rel="stylesheet" href="css/main.css">
  <link rel="icon" href="favicon.png">
  <link href="https://fonts.googleapis.com/css?family=Lato:300,400" rel="
    stylesheet">
  <title>LIVE | M.A.S.S</title>
</head>

<body>
  <div class="mobile-nav">
    <div class="hamburger">
      <div class="bar"></div>
      <div class="bar"></div>
      <div class="bar"></div>
    </div>
  </div>
  <nav class="nav">
    <a class="nav__link" href="index.html">Home</a>
    <a class="nav__link nav__link__active" href="#">Live</a>
    <a class="nav__link" href="recordings.html">Recordings</a>
    <a class="nav__link" href="help.html">Help</a>
  </nav>
```

```

<div id="page">
  <div class="cameras">
    <h1>Please add a camera</h1>
    <div class="cameras__links">
      <!-- <a class="cameras__links__link" href="#">Garden</a> -->
      <div class="cameras__links__actions">
        <!-- <a class="cameras__links__link cameras__links__actions__action"
href="#">Next</a> -->
      </div>
    </div>
  </div>
  <div class="camera-video">
    <!-- <img class="camera-img" src="" alt="camera frame" /> -->
  </div>
  <!-- <p class="camera-ip"></p> -->
  <div class="camera-actions">

    <a id="btn-add" class="camera-actions__action button button--add">Add
camera</a>
    <div class="btn-add-window">
      <p id="btn-add-window-info">a</p>
      <div class="btn-add-window__progress-bar">
        <div class="btn-add-window__progress-bar__progress"></div>
      </div>
    </div>
  </div>
  <div class="notice">
    <h2>M.A.S.S</h2>
    <p>M.A.S.S (or Machine-Assisted Surveillance System) is a Surveillance
system that incorporates Machine Learning to
      recognise faces that is easy to use and deploy.</p>
    <p>The goal of this project is to create a responsive web-site that will be
easy-to-use across multiple devices,
      including Desktops, Mobiles, and Smart Watches.</p>
    <p>Source code (soon to be) available at: <a href="https://github.com/
jevangit">https://github.com/jevangit</a></p>
  </div>
</div>
<script src="js/main.js"></script>
</body>

</html>

```

Appendix C

“Recordings” Page

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="description" content="Easy Machine-Assisted Surveillance">
  <meta name="theme-color" content="#3c3c3c" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" href="favicon.png">
  <link rel="stylesheet" href="css/main.css">
  <link rel="icon" href="favicon.png">
  <link href="https://fonts.googleapis.com/css?family=Lato:300,400" rel="
    stylesheet">
  <title>Recordings | M.A.S.S</title>
</head>

<body>
  <div class="mobile-nav">
    <div class="hamburger">
      <div class="bar"></div>
      <div class="bar"></div>
      <div class="bar"></div>
    </div>
  </div>
  <nav class="nav">
    <a class="nav__link" href="index.html">Home</a>
    <a class="nav__link" href="live.html">Live</a>
    <a class="nav__link nav__link__active" href="#">Recordings</a>
    <a class="nav__link" href="help.html">Help</a>
  </nav>
```

```

<div id="page">
  <h1>Recordings</h1>
  <p>M.A.S.S automatically stores recordings of suspected malicious activity
  for you to review.</p>
  <p>Please review the incidents below.</p>

  <table class="recordings">
    <tr>
      <th>No.</th>
      <th>Date and Time</th>
      <th>Duration</th>
    </tr>
    <tr class="can-collapse">
      <td>1</td>
      <td>2018-11-26 19:21</td>
      <td>14 seconds</td>
    </tr>
    <tr class="can-collapse-content">
      <td colspan="4" class="row-info">
        
        <div class="can-collapse-content-text">
          <p class="recording-destination">Porch</p>
          <p>eqhqehqehtrwhwrtjhwrtjwr rtjwryj tqjhqrtj qtrjqrtj tqrjqrtj
          qtehtqhijohqjotp pojiqrhopijqthjpo kopwthrkwtrhpokwthr pkowrthpowrth
          opkwtrhopkwrth <a href="#" class="btn--edit">Edit</a></p>
        </div>
        <div class="recordings__actions">
          <a href="img/streams/1.jpg" class="button button--download">Download
        </a>
          <a href="#" class="button button--remove">Remove</a>
        </div>
      </td>
    </tr>

    <tr class="can-collapse">
      <td>2</td>
      <td>2018-11-26 20:21</td>
      <td>4 seconds</td>
    </tr>
    <tr class="can-collapse-content">
      <td colspan="4" class="row-info">
        
        <div class="can-collapse-content-text">
          <p class="recording-destination">Porch</p>
          <p>eqhqehqehtrwhwrtjhwrtjwr rtjwryj tqjhqrtj qtrjqrtj tqrjqrtj
          qtehtqhijohqjotp pojiqrhopijqthjpo kopwthrkwtrhpokwthr pkowrthpowrth
          opkwtrhopkwrth <a href="#" class="btn--edit">Edit</a>

```

```

        </div>
        <div class="recordings__actions">
            <a href="img/streams/2.jpg" class="button button--download">Download
    </a>
            <a href="#" class="button button--remove">Remove</a>
        </div>
    </td>
</tr>

<tr class="can-collapse">
    <td>3</td>
    <td>2018-10-27 17:20</td>
    <td>11 seconds</td>
</tr>
<tr class="can-collapse-content">
    <td colspan="4" class="row-info">
        
        <div class="can-collapse-content-text">
            <p class="recording-destination">Porch</p>
            eqhqehqehtrwhwrtjhwrtjwr rtjwryj tqjhqrtj qtrjqrtj tqrjqrtj
qtehthqijohqjotp pojiqrhopijqthjpo kopwthrkpwtrhpokwthr pkowrthpowrth
opkwtrhopkwtrth <a href="#" class="btn--edit">Edit</a>
        </div>
        <div class="recordings__actions">
            <a href="img/streams/3.jpg" class="button button--download">Download
    </a>
            <a href="#" class="button button--remove">Remove</a>
        </div>
    </td>
</tr>

<tr class="can-collapse">
    <td>4</td>
    <td>2018-11-26 21:21</td>
    <td>24 seconds</td>
</tr>
<tr class="can-collapse-content">
    <td colspan="4" class="row-info">
        
        <div class="can-collapse-content-text">
            <p class="recording-destination">Porch</p>
            eqhqehqehtrwhwrtjhwrtjwr rtjwryj tqjhqrtj qtrjqrtj tqrjqrtj
qtehthqijohqjotp pojiqrhopijqthjpo kopwthrkpwtrhpokwthr pkowrthpowrth
opkwtrhopkwtrth <a href="#" class="btn--edit">Edit</a>
        </div>
        <div class="recordings__actions">

```

```

        <a href="img/streams/4.jpg" class="button button--download">Download
    </a>

        <a href="#" class="button button--remove">Remove</a>
    </div>
</td>
</tr>

<tr class="can-collapse">
    <td>5</td>
    <td>2018-10-26 21:40</td>
    <td>2 minutes</td>
</tr>
<tr class="can-collapse-content">
    <td colspan="4" class="row-info">
        
        <div class="can-collapse-content-text">
            <p class="recording-destination">Porch</p>
            eqhqehqehtrwhwrtjhwrtjwr rtjwryj tqjhqrtj qtrjqrtj tqrjqrtj
qtehtqhijohqjotp pojiqrhopijqthjpo kopwthrkwtrhpokwthr pkowrthpowrth
opkwtrhopkwrth <a href="#" class="btn--edit">Edit</a>
        </div>
        <div class="recordings__actions">
            <a href="img/streams/5.jpg" class="button button--download">Download
    </a>

            <a href="#" class="button button--remove">Remove</a>
        </div>
    </td>
</tr>

<tr class="can-collapse">
    <td>6</td>
    <td>2018-11-26 20:21</td>
    <td>4 seconds</td>
</tr>
<tr class="can-collapse-content">
    <td colspan="4" class="row-info">
        
        <div class="can-collapse-content-text">
            <p class="recording-destination">Porch</p>
            eqhqehqehtrwhwrtjhwrtjwr rtjwryj tqjhqrtj qtrjqrtj tqrjqrtj
qtehtqhijohqjotp pojiqrhopijqthjpo kopwthrkwtrhpokwthr pkowrthpowrth
opkwtrhopkwrth <a href="#" class="btn--edit">Edit</a>
        </div>
        <div class="recordings__actions">
            <a href="img/streams/6.jpg" class="button button--download">Download
    </a>

```

```
        <a href="#" class="button button--remove">Remove</a>
      </div>
    </td>
  </tr>

</table>
</div>
<script src="js/main.js"></script>
</body>

</html>
```

“Recordings” Page

Appendix D

“FAQ” Page

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="description" content="Easy Machine-Assisted Surveillance">
  <meta name="theme-color" content="#3c3c3c" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" href="favicon.png">
  <link rel="stylesheet" href="css/main.css">
  <link rel="icon" href="favicon.png">
  <link href="https://fonts.googleapis.com/css?family=Lato:300,400" rel="
    stylesheet">
  <title>Help | M.A.S.S</title>
</head>

<body>
  <div class="mobile-nav">
    <div class="hamburger">
      <div class="bar"></div>
      <div class="bar"></div>
      <div class="bar"></div>
    </div>
  </div>
  <nav class="nav">
    <a class="nav__link" href="index.html">Home</a>
    <a class="nav__link" href="live.html">Live</a>
    <a class="nav__link" href="recordings.html">Recordings</a>
    <a class="nav__link nav__link__active" href="#">Help</a>
  </nav>
```

```

<div id="page">
  <h1>M.A.S.S Support</h1>
  <h2>Questions and Answers</h2>

  <table class="recordings">
    <tr class="can-collapse">
      <td>Q1: What is M.A.S.S ?</td>
    </tr>
    <tr class="can-collapse-content">
      <td>
        M.A.S.S is an acronym which stands for Machine-Assisted Surveillance
        System.
        This is a project which is aimed at creating a surveillance system
        using Responsive
        Web Design.
      </td>
    </tr>

    <tr class="can-collapse">
      <td>Q2: Why use a responsive Website for this ?</td>
    </tr>
    <tr class="can-collapse-content">
      <td>
        Responsive Web Design allows developers to easily create designs that
        are capable of adapting
        to multiple resolution through the use of <a href="https://developer.
        mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries">Media
        queries.</a>
      </td>
    </tr>

    <tr class="can-collapse">
      <td>Q3: Why doesn't this work?</td>
    </tr>
    <tr class="can-collapse-content">
      <td>
        At the moment, M.A.S.S is a project to demonstrate the use of
        responsive web design in the implementation of surveillance system, thus
        functionality such as a face-recognition is not required.
      </td>
    </tr>

  </table>
  <div>
    <h2>Contact details for support and other enquiries</h2>
    <p>Jonathan Evans</p>
    <p>Email: <a href="mailto:jevans.151713@student.uwtsd.ac.uk">jevans.151713
    @student.uwtsd.ac.uk</a></p>
  </div>

```

```
<p>GitLab: <a href="https://gitlab.com/jevansgit">https://gitlab.com/
jevansgit</a></p>
<p>GitHub: <a href="https://github.com/jevansgit">https://github.com/
jevansgit</a></p>
</div>
</div>
<script src="js/main.js"></script>
</body>

</html>
```

“FAQ (Frequently Asked Questions)” Page

Appendix E

JavaScript

```
var cameraStream = [];  
  
var btnAdd = document.querySelector('#btn-add');  
var windowAdd = document.querySelector('.btn-add-window');  
var infoBtnAdd = document.querySelector('#btn-add-window-info');  
var progressBtnAdd = document.querySelector('.btn-add-window__progress-  
    bar__progress');  
var progressBar = document.querySelector(".btn-add-window__progress-bar");  
var page = document.querySelector('#page');  
var btnHamburger = document.querySelector('.hamburger');  
var navMain = document.querySelector('nav');  
var camerasAdded = document.querySelector('.cameras__links');  
var camerasAddedLink = document.getElementsByClassName('cameras__links__link');  
var currentCameraHeading = document.querySelector('.is-live');  
var cameraIp = document.querySelector('.camera-ip');  
  
var cameras;  
var camerasFound = false;  
var blockingMenu = false;  
  
if (btnHamburger != null) {  
    btnHamburger.onclick = function () {  
        if (navMain.classList.contains("hidden")) {  
            navMain.classList.remove("hidden");  
            document.body.style.backgroundColor = "rgba(0,0,0,0.25)";  
        } else {  
            navMain.classList.add("hidden");  
        }  
    }  
}
```

```

}

var scanAlreadyRun = false;
var firstAdded = false;
if (btnAdd != null) {
    btnAdd.onclick = function () {

        if (document.querySelector('.camera-img') == null) {
            windowAdd.style.marginTop = "30px";
            windowAdd.style.marginLeft = "4px";
        } else {
            windowAdd.style.marginLeft = "0px";
        }
        let progress = 0;

        if (windowAdd.classList.contains("on")) {
            windowAdd.classList.remove("on");
            console.log("* Turning off camera discovery ... *")
        } else {
            windowAdd.classList.add("on");
            console.log("* Openning camera discovery ... *")
            var i = 1;

            if (!scanAlreadyRun) {
                (function sleep(i) {
                    setTimeout(function () {
                        scanAlreadyRun = true;
                        progress++;
                        infoBtnAdd.textContent = "Searching IP ranges for cameras
... " + progress + "%";
                        progressBtnAdd.style.width = progress + "%";
                        if (--i) sleep(i)
                    } else {
                        progressBtnAdd.style.background = "transparent";
                        progressBar.style.background = "transparent";
                        if (document.querySelector('.camera-img') == null) {
                            windowAdd.style.marginTop = "30px";
                        } else { windowAdd.style.marginTop = "-380px"; }
                        windowAdd.style.height = "350px";
                        windowAdd.style.overflowY = "auto";

                        for (let i = 0; i < 10; i++) {
                            var cameraItem = document.createElement("div");

                            var camera = document.createElement("div");
                            var cameraName = document.createTextNode
("192.168.0." + (i + 1));

```

```

        camera.appendChild(cameraName);
        camera.className = "camera";

        cameraItem.appendChild(camera);
        cameraItem.className = "camera-item";
        windowAdd.append(cameraItem);
    }

    var cameras = document.querySelectorAll('.camera-item

');

    // Add event listeners to each camera item
    for (let i = 0; i < cameras.length; i++) {
        cameras[i].addEventListener('click', function(
event) {

            console.log(i);
            var txtAdd = document.createElement("input");
            txtAdd.setAttribute("type", "text");
            txtAdd.className = "txtCamera";
            txtAdd.setAttribute("placeholder", "Enter

camera name here...");

            var btnSave = document.createElement("div");
            btnSave.className = "btnSave"
            btnSaveName = document.createTextNode("Add");
            btnSave.append(btnSaveName);

            txtAdd.addEventListener('keyup', function(
event) {

                if (event.keyCode === 13) {
                    btnSave.click();
                }
            })

            btnSave.addEventListener('click', function(
event) {

                console.log("Saving " + txtAdd.value);
                cameras[i].textContent = txtAdd.value;
                cameras[i].style.color = "#ccc";
                cameras[i].style.cursor = "default";

                var notification = document.createElement
("div");

                var notificationText = document.
createTextNode(txtAdd.value + " camera has been added");
                notification.appendChild(notificationText

);

                notification.className = "notification

notification--add";

```

```

1) + ".jpg";

document.body.appendChild(notification);
var currentImage = "img/streams/" + (i +

cameraStream.push ({
    ip: "192.168.0." + (i + 1),
    name: txtAdd.value,
    imageSrc: currentImage
});

console.log(cameraStream);

txtAdd.remove();
btnSave.remove();

let existingImage = document.
querySelector('.camera-img');

var cameraVideo = document.querySelector
('.camera-video');

if (existingImage === null) {
    let camActions = document.
        querySelector('.camera-actions');

    camActions.style.padding = "0";
    camActions.style.border = "none"
    camActions.style.textAlign = "left";

    windowAdd.style.marginTop = "-450px";
    console.log("Image not found, adding

});

    }
    refreshCameras();
});

// Use insertBefore to add textbox after a
given camera item

let parentDiv = cameras[i].parentNode;
parentDiv.insertBefore(txtAdd, cameras[i +

parentDiv.insertBefore(btnSave, cameras[i +

1]);

1]);

});
}

```

```

        blockingMenu = true;

        infoBtnAdd.textContent = "Found 10 cameras...";
        var btnClose = document.createElement("a");
        var btnCloseText = document.createTextNode("Close");
        btnClose.appendChild(btnCloseText);
        btnClose.className = "button button--close";
        btnClose.style.display = "block";
        windowAdd.append(btnClose);

        btnClose.onclick = function () {
            windowAdd.classList.remove("on");
        }

        };
    }, 10);
})(100);
}
}
windowAdd.style.marginLeft = "0px";
}
}

if (page != null)
    page.onclick = function () {
        // if (windowAdd != null){
            // if (windowAdd.classList.contains("on") && blockingMenu == false) {
                //     windowAdd.classList.remove("on");
            // }
        // }

        if (!navMain.classList.contains("hidden") && window.innerWidth <= 768) {
            navMain.classList.add("hidden");
            document.body.style.backgroundColor = "#fefefe";
        }
    }
}

// Disabling this menu JavaScript allows for non-javascript devices to use mobile
// navigation
document.getElementsByTagName("body")[0].onresize = function () {
    updateNav()
};

function updateNav() {
    if (window.innerWidth <= 768) {
        navMain.classList.add("hidden");
    } else {
        if (navMain.classList.contains("hidden")) {
            navMain.classList.remove("hidden");
        }
    }
}

```

```

    }
}
updateNav();

function refreshCameras() {
    console.log("[Cameras]: Refreshing...");
    console.log("[Cameras]: Object array length: " + cameraStream.length);
    while (camerasAddedLink[0]) {
        if (camerasAddedLink[0].parentNode != null) {
            camerasAddedLink[0].parentNode.removeChild(camerasAddedLink[0]);
            console.log("[Cameras]: Clearing...");
        } else { console.log("Can't find parent node"); }
    }

    if (cameraStream.length > 0 && document.querySelector('#btn-remove') === null) {
        btnRemove = document.createElement("a");
        let actions = document.querySelector('.camera-actions');
        actions.style.float = "right";
        btnRemove.id = "btn-remove";
        btnRemove.className = "camera-actions__action button button--remove button--rounded";
        btnRemove.textContent = "Remove";

        btnRemove.addEventListener('click', function (event) {
            cameraStream.shift();

            if (cameraStream.length == 0) {
                document.querySelector('#btn-remove').remove();
                document.querySelector('h1').textContent = "Please add a camera";
                document.querySelector('h1').classList.remove("is-live");
                document.querySelector('img').remove();
                document.querySelector('.camera-ip').remove();
            }

            refreshCameras();
        })
        actions.appendChild(btnRemove);
    } else {

        if (cameraStream.length === 0) {
            let camActions = document.querySelector('.camera-actions');
            camActions.style.padding = "0";
            camActions.style.border = "none";
            camActions.style.textAlign = "left";

            windowAdd.style.marginTop = "-450px";

```

```

        let actions = document.querySelector('.camera-actions');
        actions.style.float = "left";
        actions.style.margin = "20px 0";
        actions.style.padding = "";
        actions.style.float = "";
    }
}

if (cameraStream.length > 0) {
    document.querySelector('h1').textContent = cameraStream[0].name;
    document.querySelector('h1').classList.add("is-live");
    let currentImage = document.querySelector('img');

    if (currentImage !== null) {
        currentImage.src = cameraStream[0].imageSrc;

    } else {
        var image = document.createElement("img");
        image.src = cameraStream[0].imageSrc;

        document.querySelector('.camera-video').appendChild(image);
    }

    let ip = document.querySelector('.camera-ip');
    if (ip !== null) {
        ip.textContent = cameraStream[0].ip;
    } else {
        let newIp = document.createElement("p");
        newIp.classList.add("camera-ip");
        newIp.textContent = cameraStream[0].ip;

        let video = document.querySelector('.camera-video');
        video.parentNode.insertBefore(newIp, video.nextSibling);
    }
}

if (camerasAdded !== null) {
    console.log(cameraStream);

    for (let i = 0; i < cameraStream.length; i++) {

        var newCamera = document.createElement("a");
        var newCameraText = document.createTextNode(cameraStream[i].name);
        newCamera.className = "cameras__links__link";
        newCamera.appendChild(newCameraText);
        newCamera.addEventListener('click', function () {

            // Swap with front of queue

```

```

        console.log("clicked " + cameraStream[i].name);
        let tmpStream = cameraStream[i];
        cameraStream[i] = cameraStream[0];
        cameraStream[0] = tmpStream;

        console.log("222222");
        console.log(cameraStream);

        console.log("Viewing camera: " + cameraStream[i].name);

        refreshCameras();
    });

    if (cameraStream[0] !== cameraStream[i]) {
        camerasAdded.appendChild(newCamera);
    }
}

if (cameraStream.length > 4) {
    var newBtn = document.createElement("a");
    var newBtnText = document.createTextNode("Next");
    newBtn.className = "cameras__links__link
cameras__links__actions__action";
    newBtn.appendChild(newBtnText);

    camerasAdded.appendChild(newBtn);
}
}

/*
 * Recordings page JS
 */
var recordings = {
    recording1 : {
        No: 1,
        DateTime: "2018-11-26 19:21",
        Duration: "14 seconds",
        Description: "eqhqehqehtrwhrtjhwrtjwr rtjwryj tqjhqrtj qtrjqrtj tqrjqrtj
qtehtqijohqjotp pojiqrhopijqthjpo kopwthrkpwrthpokwthr pkowrthpowrth
opkwtrhopkwrth Lorem ipsum",
        ImageSrc: "img/streams/1.jpg",
    },
    recording2 : {
        No: 2,
        DateTime: "2018-11-26 19:21",
        Duration: "24 seconds",
    }
}

```

```

        Description: "eqhqehqehtrwhwrtjhwrtjwr rtjwryj tqjhqrtj qtrjqrtj tqrjqrtj
        qtehthqijohqjotp pojiqrhopijqthjpo kopwthrkpwtrhpokwthr pkowrthpowrth
        opkwtrhopkwrth Lorem ipsum",
        ImageSrc: "img/streams/2.jpg",
    },
    recording3 : {
        No: 3,
        DateTime: "2018-11-26 19:21",
        Duration: "15 seconds",
        Description: "eqhqehqehtrwhwrtjhwrtjwr rtjwryj tqjhqrtj qtrjqrtj tqrjqrtj
        qtehthqijohqjotp pojiqrhopijqthjpo kopwthrkpwtrhpokwthr pkowrthpowrth
        opkwtrhopkwrth Lorem ipsum",
        ImageSrc: "img/streams/3.jpg",
    },
    recording4 : {
        No: 4,
        DateTime: "2018-11-26 19:21",
        Duration: "4 seconds",
        Description: "eqhqehqehtrwhwrtjhwrtjwr rtjwryj tqjhqrtj qtrjqrtj tqrjqrtj
        qtehthqijohqjotp pojiqrhopijqthjpo kopwthrkpwtrhpokwthr pkowrthpowrth
        opkwtrhopkwrth Lorem ipsum",
        ImageSrc: "img/streams/4.jpg",
    },
}

var recordingArray = [];
for (var i in recordings) {
    recordingArray.push(recordings[i]);
}

var recordingsTable = document.querySelector('.recordings');

/*
 * Load data
 */
/*
if (recordingsTable != null) {
    for (let i = 0; i < recordingArray.length; i++) {
        console.log("loading record: " + i);
        let tr = document.createElement("tr");

        let tdId = document.createElement("td");
        tdId.textContent = recordingArray[i].No;

        let tdDateTime = document.createElement("td");
        tdDateTime.textContent = recordingArray[i].DateTime;
    }
}

```

```

        let tdDuration = document.createElement("td");
        tdDuration.textContent = recordingArray[i].Duration;

        tr.appendChild(tdId);
        tr.appendChild(tdDateTime);
        tr.appendChild(tdDuration);

        let tr2 = document.createElement("tr");

        tr.addEventListener('click', function (event) {
            let expanded = 0;
            console.log("expanding");
            if (expanded == 0) {
                expanded = 1;

                let trExpand = document.createElement("tr");
                let tdRowInfo = document.createElement("td");

                tdRowInfo.className = "row-info";
                tdRowInfo.colSpan = 4;
                tdRowInfo.innerHTML = "<img src=\"" + recordingArray[i].ImageSrc
+ "\" />";
                tdRowInfo.innerHTML += "<p>" + recordingArray[i].Description + "
</p>";
                tdRowInfo.innerHTML += "<a class=\""button button--add\" href
=\"#">Download</a>";
                tdRowInfo.innerHTML += "<a class=\""button button--remove\" href
=\"#">Remove</a>";

                trExpand.appendChild(tdRowInfo);

                // recordingsTable.appendChild(trExpand);
                console.log(i);
                tr2.appendChild(trExpand);
                // tr.parentNode.appendChild(trExpand);

            } else { console.log("collapse"); }
        });

        recordingsTable.appendChild(tr);
        recordingsTable.parentNode.appendChild(tr2);
    }
}*/

var canCollapse = document.getElementsByClassName('can-collapse');
var canCollapseEdit = document.getElementsByClassName('btn--edit');
var canCollapseContent = document.getElementsByClassName('can-collapse-content');
```

```

var canCollapseContentText = document.getElementsByClassName('can-collapse-
    content-text');
var canCollapseRemove = document.getElementsByClassName('button--remove');
console.log(canCollapseEdit.length);
for (let i = 0; i < canCollapse.length; i++) {
    canCollapse[i].nextElementSibling.style.display = "none";

    canCollapse[i].addEventListener('click', function (event) {
        console.log("clicked");
        var content = this.nextElementSibling;

        if (content.style.display === "") {
            content.style.display = "none";
            canCollapse[i].style.background="#eee url('img/expand-icon.png') no-
repeat 97% 50%";
        } else {
            content.style.display = "";
            canCollapse[i].style.background="#fff url('img/collapse-icon.png') no
-repeat 97% 50%";

        }
    });

    if (canCollapseEdit[i] != null) {
        canCollapseEdit[i].addEventListener('click', function (event) {
            var txtInput = prompt ("Please enter text here:");
            if (txtInput != null || txtInput != "") {
                canCollapseContentText[i].textContent = txtInput;
            } else { console.log("Invalid input!"); }
        });
    }

    if (canCollapseRemove[i] != null) {
        canCollapseRemove[i].addEventListener('click', function (event) {
            canCollapse[i].remove();
            canCollapseContentText[i].remove();
            canCollapseContent[i].remove();

            var notification = document.createElement("div");
            var notificationText = document.createTextNode("Removed recording");
            notification.appendChild(notificationText);
            notification.className = "notification--remove notification";
            document.body.appendChild(notification);

        });
    }
}
}

```

JavaScript code

Appendix F

Log book

See proceeding pages for log book entries

Entry #1

PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David

☐ RESEARCH
☐ PROJECT
☐ DISSERTATION
☐ MENTORING

SUPERVISION FORM

Degree Title: BSc SW Eng

Supervisor: Mikie Dacey

Name of Student: Jonathan Evans

Supervision Meeting Date: 25/10/18

Length: 20m

- Objective of Meeting:

Review Progress

- Work Submitted by Student:

- Issues Discussed at Meeting:

Discussed changing topic to usability

- Student Action BEFORE Next Meeting:

Lit review - start writing first few pages
Design some interfaces

Date of Next Meeting: 1/11/18

Signed:(Student) Jonathan Evans

Signed:(Supervisor) M. Dacey

Top copy to student • 2nd copy to Co-ordinator • 3rd copy retain for your records

Entry #2

 PRIFYSGOL CYMRU Y Drindod Dewi Sant UNIVERSITY OF WALES Trinity Saint David	☐ RESEARCH ☐ PROJECT ☐ DISSERTATION ☐ MENTORING	SUPERVISION FORM
--	--	--

Degree Title: <u>BSc Sw Eng</u>	Supervisor: <u>MISS DACEY</u>
Name of Student: <u>Jonathan Evans</u>	
Supervision Meeting Date: <u>1 / 11 / 18</u>	Length: <u>20 m</u>

• Objective of Meeting:

Review Progress

• Work Submitted by Student:

- demo web design
- small amount of lit review

• Issues Discussed at Meeting:

Cross platform user interface design for surveillance → responsive web design

• Student Action BEFORE Next Meeting:

- Produce project title to reflect usability, responsive web design and surveillance applications.
- 5 pages of lit review
- User experience → write the research methodology section first

Date of Next Meeting: 6 / 1 / 18

Signed: (Student) <u>Jonathan Evans</u>	Signed: (Supervisor) <u>M. E. Dacey</u>
---	---

Top copy to student • 2nd copy to Co-ordinator • 3rd copy retain for your records

Entry #3

 PRIFYSGOL CYMRU Y Drindod Dewi Sant UNIVERSITY OF WALES Trinity Saint David	☐ RESEARCH ☐ PROJECT ☐ DISSERTATION ☐ MENTORING	SUPERVISION FORM
--	--	-----------------------------

Degree Title: <u>BSc Software Eng.</u>	Supervisor: <u>MRS DACEY</u>
Name of Student: <u>Jonathan Evans</u>	
Supervision Meeting Date: <u>6/11/18</u>	Length: <u>20</u>

• Objective of Meeting: Review start of literature review.

• Work Submitted by Student: Initial lit review.

• Issues Discussed at Meeting: Progr. - last development of an interface for prototype 1.

• Student Action BEFORE Next Meeting:

- Initial implementation of prototype
- Add more content to lit review.

Date of Next Meeting: 15/11/18

Signed:(Student) Jonathan Evans Signed:(Supervisor) M.E. Dacey

Top copy to student • 2nd copy to Co-ordinator • 3rd copy retain for your records

Entry #4

 Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David

☐ RESEARCH
☐ PROJECT
☐ DISSERTATION
☐ MENTORING

SUPERVISION FORM

Degree Title: Software Engineering Supervisor: Mike Dacey
Name of Student: Jonathan Evans
Supervision Meeting Date: 15 / 11 / 18 Length: 20

• Objective of Meeting:
Review progress.

• Work Submitted by Student:
Code submitted to GIT. Brief implementation of main page.
- HTML / SAS + GULP
- Adding JS for camera switching

• Issues Discussed at Meeting:
Make sure to consider the usability tests you wish to use.

• Student Action BEFORE Next Meeting:
- Complete first prototype ready for testing.
- Video your tests.

Date of Next Meeting: 22 / 11 / 18

Signed:(Student) Jonathan Evans Signed:(Supervisor) M. Dacey

Top copy to student • 2nd copy to Co-ordinator • 3rd copy retain for your records

Entry #5

	PRIFYSGOL CYMRU Y Drindod Dewi Sant UNIVERSITY OF WALES Trinity Saint David	☐ RESEARCH ☐ PROJECT ☐ DISSERTATION ☐ MENTORING	SUPERVISION FORM

Degree Title: Software Engineering Supervisor: Mike Dacey

Name of Student: Jonathan Evans

Supervision Meeting Date: 22/11/18 Length: 20

• Objective of Meeting:

Review page

• Work Submitted by Student:

Basic responsive web site - in firefox.

• Issues Discussed at Meeting:

- usability testing - identify tasks for testing

• Student Action BEFORE Next Meeting:

- Work on completing user interface & mobile responsiveness.

Date of Next Meeting: 29/11/2018

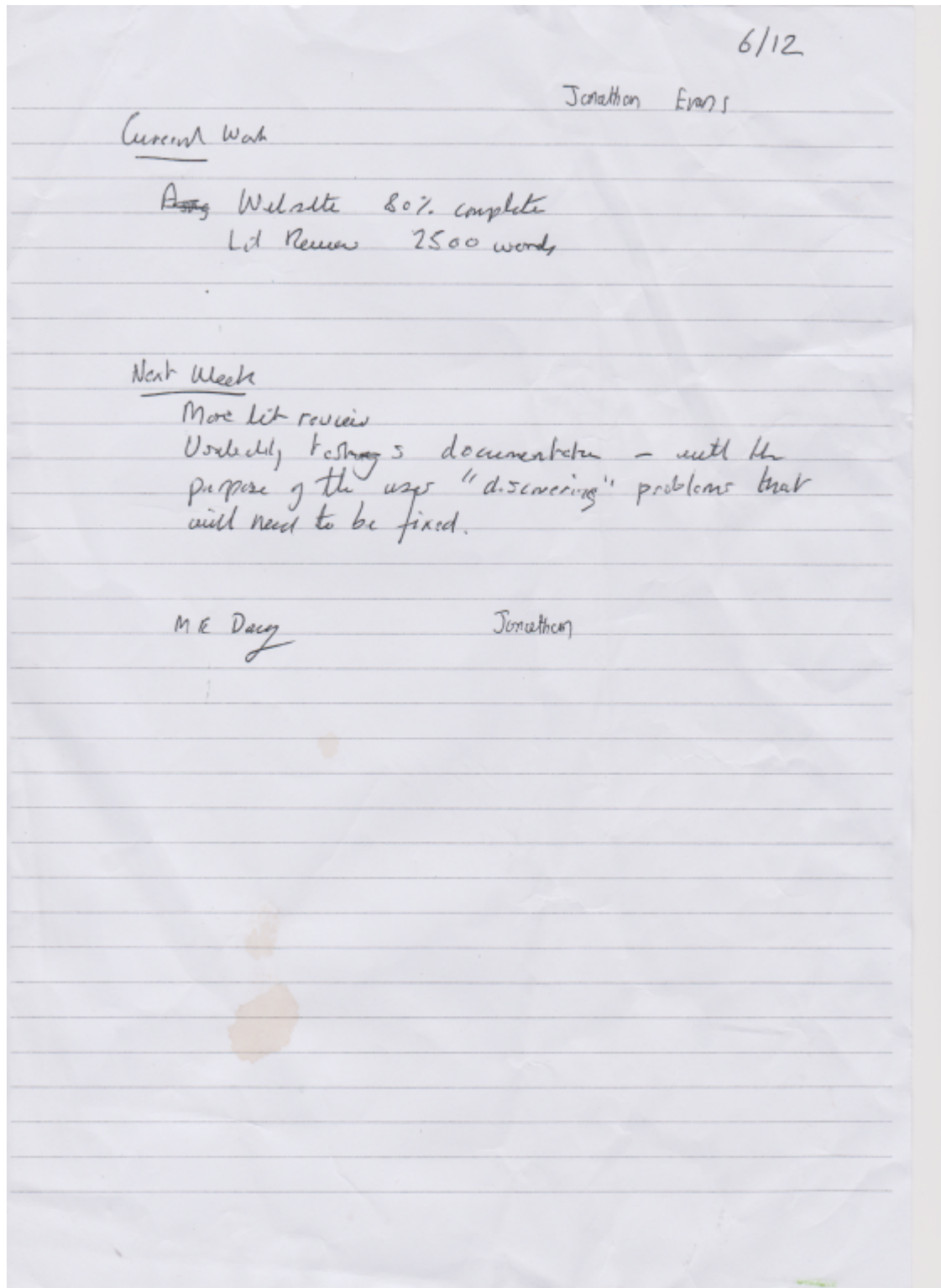
Signed:(Student) Jonathan Evans Signed:(Supervisor) Mike Dacey

Top copy to student • 2nd copy to Co-ordinator • 3rd copy retain for your records

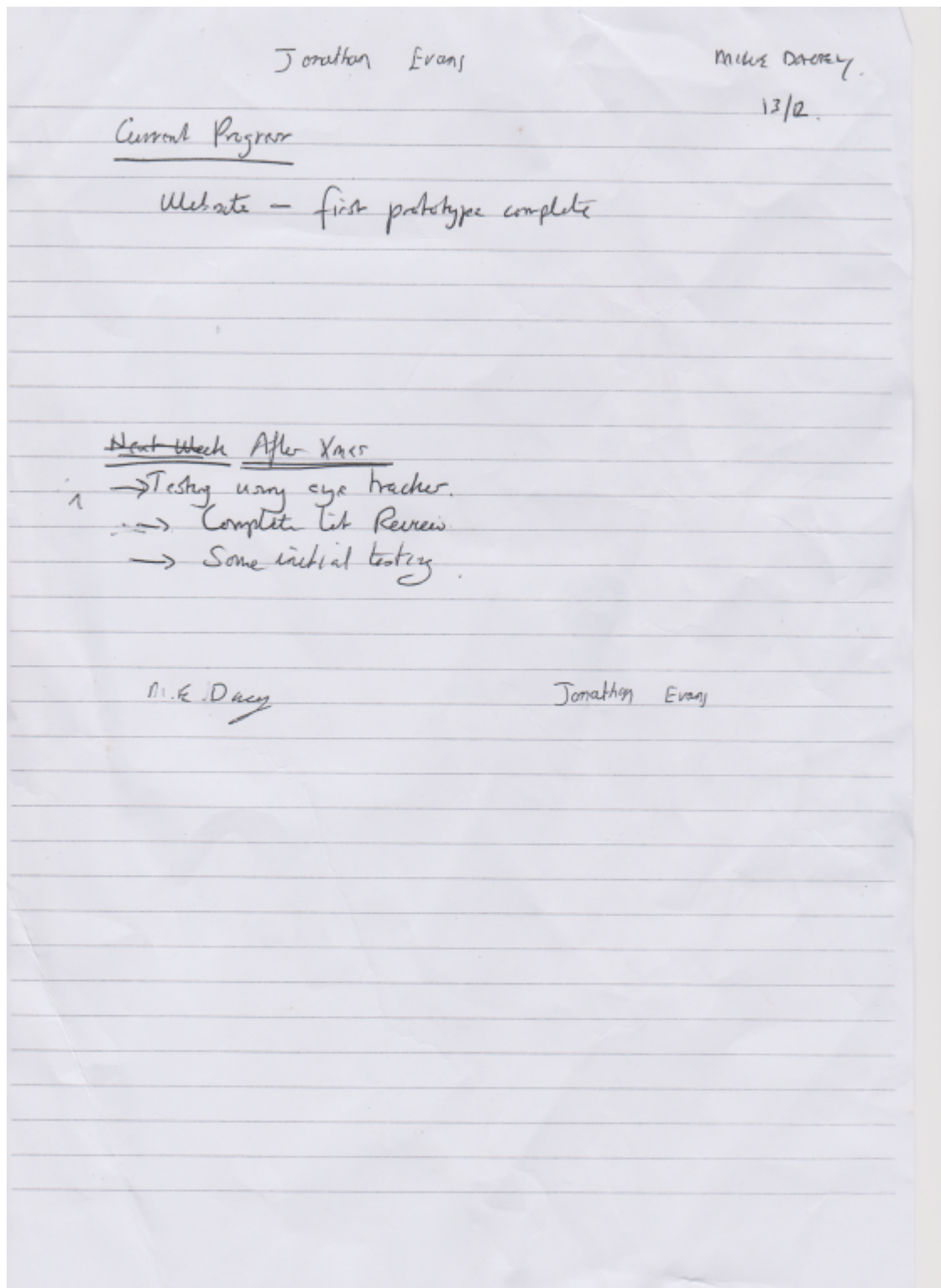
Entry #6

JONATHAN EVANS	5/2
<u>Current Progress</u>	
One usability test conducted, with eye tracker	
<u>Next week</u>	
- install eye tracking software in 204.	
- More people testing	
- Write up results of the testing	
M.E. Dancy	Jonathan Evans

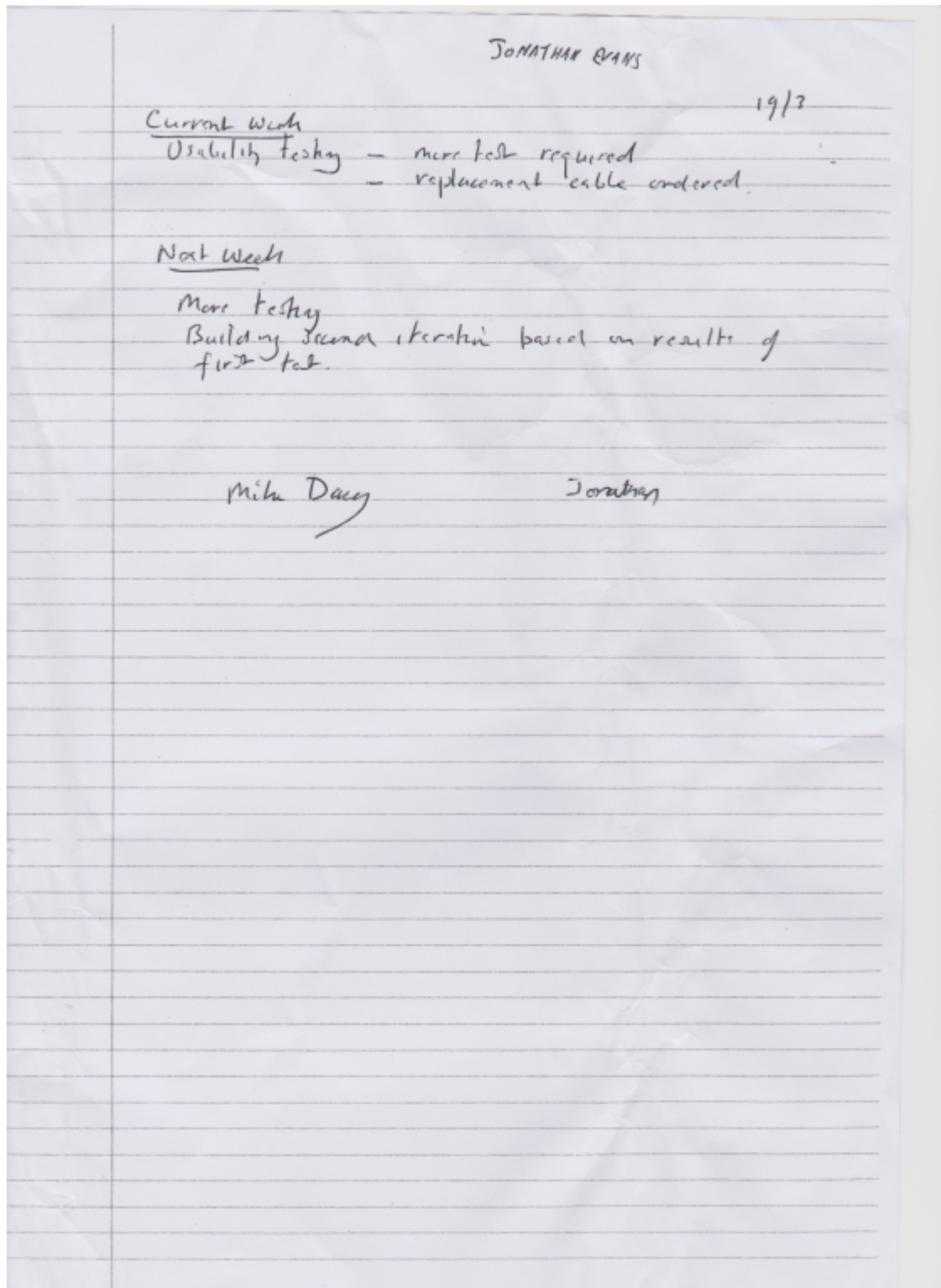
Entry #7



Entry #8



Entry #9



Log book

Bibliography

- [1] StatCounter . Mobile and tablet internet usage exceeds desktop for first time worldwide, . URL <http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide>.
- [2] Christian Rohrer. When to Use Which User-Experience Research Methods. URL <https://www.nngroup.com/articles/which-ux-research-methods/>.
- [3] Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '13*, page 1168. ACM Press, . ISBN 978-1-4503-2174-7. doi: 10.1145/2487575.2488217. URL <http://dl.acm.org/citation.cfm?doid=2487575.2488217>.
- [4] Ron Kohavi, Randal M. Henne, and Dan Sommerfield. Practical guide to controlled experiments on the web: Listen to your customers not to the hippo. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '07*, page 959. ACM Press, . ISBN 978-1-59593-609-7. doi: 10.1145/1281192.1281295. URL <http://portal.acm.org/citation.cfm?doid=1281192.1281295>.
- [5] Aurora Harley. Multivariate vs. A/B Testing: Incremental vs. Radical Changes. URL <https://www.nngroup.com/articles/multivariate-testing/>.
- [6] Obama 2012. URL <https://www.optimizely.com/customers/obama2012/>.
- [7] Task Scenarios for Usability Testing. URL <https://www.nngroup.com/articles/task-scenarios-usability-testing/>.
- [8] Jakob Nielsen. Thinking Aloud. In *Usability Engineering*, pages 115–148. Morgan Kaufmann, . ISBN 0-12-518406-9.

- [9] K. Anders Ericsson and Herbert A. Simon. Verbal reports as data. 87(3):215–251. ISSN 0033-295X. doi: 10.1037/0033-295X.87.3.215. URL <http://content.apa.org/journals/rev/87/3/215>.
- [10] OGAMA (OpenGazeAndMouseAnalyzer): An open source software designed to analyze eye and mouse movements in slideshow study designs. URL <http://www.ogama.net/>.
- [11] Why You Only Need to Test with 5 Users. URL <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
- [12] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Empowering People - CHI '90*, pages 249–256. ACM Press. ISBN 978-0-201-50932-8. doi: 10.1145/97243.97281. URL <http://portal.acm.org/citation.cfm?doid=97243.97281>.
- [13] Jakob Nielsen. Usability Heuristics. In *Usability Engineering*, pages 115–148. Morgan Kaufmann, . ISBN 0-12-518406-9.
- [14] StatCounter . Operating System Market Share Worldwide, . URL <http://gs.statcounter.com/os-market-share>.
- [15] Donald A. Norman. *The Design of Everyday Things*. Basic Books, revised and expanded edition edition. ISBN 978-0-465-05065-9.
- [16] W3C . Web Content Accessibility Guidelines (WCAG) 2.0, . URL <https://www.w3.org/TR/WCAG20/>.
- [17] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer-Interaction*. Addison Wesley Longman, 3rd ed edition. ISBN 978-0-201-69497-0.
- [18] Open Web Platform Milestone Achieved with HTML5 Recommendation. URL <https://www.w3.org/2014/10/html5-rec.html.en>.
- [19] Virtual DOM and Internals – React. URL <https://reactjs.org/docs/faq-internals.html>.

-
- [20] Introduction to CSS layout. URL https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Introduction.
 - [21] CSS Flexible Box Layout Module Level 1, . URL <https://www.w3.org/TR/css-flexbox-1/>.
 - [22] CSS Grid Layout Module Level 1, . URL <https://www.w3.org/TR/css-grid-1/>.
 - [23] CSS Grid Layout Support, . URL <https://caniuse.com/#feat=css-grid>.
 - [24] CSS Flexible Box Layout Support, . URL <https://caniuse.com/#feat=flexbox>.
 - [25] World Wide Consortium (W3C). Media Queries. URL <https://www.w3.org/TR/css3-mediaqueries/>.
 - [26] iSpyConnect. iSpy - Open source camera security software. URL <http://www.ispyconnect.com>.
 - [27] Nest. Nest Cam — Indoor and outdoor home security cameras. URL <https://www.nest.com/uk/cameras/>.