



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

DESIGN AND DEVELOPMENT OF SECURITY GUARD MANAGEMENT APPLICATION FOR SMEs IN THE RETAIL INDUSTRY

by
Leonid Yanchev

Supervisor: Dr. Dr Amit Naik
Second Marker: TBD

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

December 2025

Declaration of Originality

I, **LEONID YANCHEV** 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 : LEONID YANCHEV

Date : 15-Dec-2025



FORM OF CONSENT

I **LEONID YANCHEV** hereby consent that my Project, submitted in candidature for the B.Sc.(Hons) 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 organizations.

Signed : LEONID YANCHEV

Date : 15-Dec-2025



Copyright Acknowledgement

I acknowledge that the copyright of this project report, and any product developed as part of the project, belongs to University of Wales Trinity Saint David, Swansea.

Signed: **LEONID YANCHEV**

Date : 15-Dec-2025



TABLE OF CONTENTS

ABSTRACT	6
CHAPTER 1 INTRODUCTION	8
1.1 Aim and Objectives of the Project	8
CHAPTER 2 - RESEARCH AND LITERATURE REVIEW	9
2.1 Introduction	9
2.2 Security Operations Management Systems.....	9
2.3 Mobile Application Development in Computing	10
2.4 No-Code and Low-Code Platforms.....	10
2.5 Automation and Cloud Integration in Application Development	10
2.6 Summary and Research Gaps	11
CHAPTER 3 - RESEARCH AND/OR DEVELOPMENT METHODOLOGY	12
3.1 Introduction	12
3.2 Research Philosophy	12
3.3 Research Approach	13
3.4 Research Strategy	13
3.5 Research Design - The Research Onion Framework	14
3.6 Data Collection Method	14
3.7 Evaluation Criteria.....	15
3.8 Data Analysis Methods	15
3.8.1 Quantitative Data Analysis	15
3.8.3 Qualitative Data Analysis	16
3.9 Development Methodology.....	17
3.10 Software, Tools and Hardware	18
3.11 Validity, Reliability, and Generalizability	18
3.12 Ethical Considerations	18
3.12 Chapter Summary	19
CHAPTER 4 – DESIGN & IMPLEMENTATION	20
4.1 Introduction	20
4.2 Requirement Analysis and Design Starting Point	20
4.2.1 Discovery and Refinement of Requirements	20

4.2.2 Functional and Non-Functional Requirements.....	21
4.3 Architectural Design and Technology Selection	21
4.3.1 Architectural Overview	21
4.3.2 Design Trade-Offs and Alternatives.....	22
4.4 Data Design and Information Flow	22
4.4.1 Data Structures and Schema Design	22
4.4.2 Data Synchronization Process	23
4.5 System Modelling and Process Representation	23
4.5.1 Use Case Modelling	23
4.5.2 Process Modelling.....	23
4.6 User Interface and Experience Design	23
4.6.1 Mobile Application Design	23
4.6.2 Web-Based Admin Dashboard Design	24
4.7 Implementation Strategy and Development Process	24
4.7.1 Tools and Techniques.....	24
4.7.2 Challenges Encountered and Solutions.....	25
4.8 Data Flow Overview	25
4.9 Testing and Validation Strategy.....	25
4.10 Iterative Development Overview.....	26
4.11 Chapter Summary.....	26
CHAPTER 5 –TESTING & EVALUATION	27
5.1 Introduction	27
5.2 Testing Strategy Overview	27
5.3 Unit Testing.....	28
5.4 System Testing	28
5.4 Usability Testing.....	29
5.4 Mobile Application Usability.....	29
5.5 Web Admin Dashboard Usability.....	29
5.6 Acceptance Testing.....	30
5.7 Evaluation Against Objectives	30
5.8 Evaluation Against the Project Aim	31
5.9 Comparison with Literature Review Findings.....	31

5.10 Unexpected Outcomes and Limitations	31
5.11 Chapter summary.....	31
CHAPTER 6 – CONCLUSIONS AND RECOMMENDATIONS	32
6.1 Introduction	32
6.2 Summary of the Project.....	32
6.3 Achievement of Aims and Objectives	32
6.4 Contribution to Knowledge and Practice.....	33
6.5 Benefits of the Implemented System.....	34
6.6 Limitations and Unresolved Issues	34
6.7 Recommendations and Suggestions for Further Work	35
6.7.1 Technical Enhancements	35
6.8 Chapter summary.....	36
CHAPTER 7 – REFLECTION	37
7.1 Introduction	37
7.2 Reflection on Learning and Skill Development	37
7.3 Linking Theory to Practice.....	38
7.4 Reflection on Design and Decision-Making	38
7.5 Reflection on Problem-Solving and Adaptability	39
7.6 Personal and Professional Development.....	39
7.7 Reflection on What Could Have Been Done Better	40
7.8 Learning for Future Projects.....	40
7.9 Chapter summary.....	40
REFERENCES	41
Appendix A – Mobile Application Screens (Adalo).....	44
Appendix B – Admin Web Dashboard Screens (Adalo)	46
Appendix C – Database Design & Collections (Adalo).....	49
Appendix-D – Integration & Automation Evidence	50
Appendix E – Data Flow Overview Diagram (Logical Representation)	51
Appendix F – Project Timeline (GANTT Chart)	52

ABSTRACT

Retail security operations have traditionally relied on manual, paper-based processes for recording and managing incidents such as shoplifting. While familiar and low-cost, these approaches often result in inconsistent data capture, delayed reporting, limited accessibility, and reduced management oversight. In an era where data-driven decision-making is increasingly important, such limitations can negatively impact loss prevention strategies, operational efficiency, and compliance requirements. This project addresses these challenges through the design and implementation of a digital Security Guard Management System, developed within the context of a multi-store retail environment.

The primary aim of this project was to design and implement a centralized digital solution that improves the recording, monitoring, and management of shoplifting incidents. To achieve this aim, a set of specific objectives was defined, including the digitization of incident reporting, the centralization of incident data, the improvement of data consistency and auditability, and the provision of analytical and reporting tools for management. The project also sought to explore the suitability of no-code development platforms for building complex, role-based systems within applied computing.

A development and experimentation-based methodology was adopted. The project employed a combination of requirements analysis, literature review, system design, iterative implementation, and structured testing and evaluation. The solution was implemented using a no-code mobile application builder for operational data capture by security guards, a separate web-based administrative interface for management oversight, and an integration layer using automation tools to synchronize data between systems. This architecture was chosen to balance technical feasibility, scalability, and alignment with the project's constraints.

The major deliverables of the project include a functional mobile application for security officers, a web-based admin dashboard for monitoring and managing incidents, a centralized data integration mechanism, and automated reporting capabilities, including PDF export of incident records. The system supports structured incident reporting, photo evidence attachment, role-based access control, dashboard statistics, and audit-friendly documentation. Testing confirmed that the system operates reliably across its components, with successful end-to-end data flow from incident capture to administrative review.

Evaluation of the system demonstrates that the project met its stated objectives and achieved its overarching aim. The digital solution significantly improves upon traditional manual reporting methods by enhancing data quality, accessibility, and management visibility. Findings from the project align closely with existing academic and professional literature, which suggests that digital incident management systems offer substantial benefits over paper-based approaches in terms of efficiency and decision support. While

some limitations were identified, such as reliance on third-party automation services and lack of offline functionality, these were considered acceptable within the scope and constraints of the project.

In conclusion, this project demonstrates that no-code platforms, when supported by sound architectural planning and engineering principles, can be effectively used to develop robust and scalable security management systems. The work contributes practical insight into the application of applied computing techniques within a real-world organizational context. Recommendations for future work include the integration of a dedicated backend for real-time synchronization, enhanced analytics for predictive loss prevention, and expanded usability testing. Overall, the project provides a strong foundation for further research and development in digital security management solutions.

CHAPTER 1 INTRODUCTION

In today's fast-paced retail and corporate environment, the need for structured, technology-driven security management systems has become increasingly important. Many organizations, including major retail chains, employ security personnel to ensure the safety of staff, customers, and assets. However, many these operations still rely on manual methods of record-keeping—such as paper-based attendance logs, handwritten incident reports, and daily checklists. These traditional practices are often inefficient, prone to human error, and difficult to monitor across multiple locations. The lack of a centralized system for data collection and analysis not only limits managerial oversight but also hinders effective decision-making and accountability [1],[2].

To address this issue, this project focuses on the design and development of a Guard Management Mobile Application using Adalo, a no-code development platform that enables rapid creation of mobile apps without programming expertise [3]. The app aims to streamline essential security operations, including daily activity checklists, clock-in/clock-out attendance logging, and automated timesheet generation for payroll processing. By integrating Google Sheets for data storage and Zapier for automation, the system provides a cost-effective, scalable, and user-friendly solution that can be implemented in retail and corporate environments such as Iron Chevron Limited [4],[5].

1.1 Aim and Objectives of the Project

The aim of this project is to explore whether no-code technologies can effectively support the development of scalable solution for managing and monitoring security guard activities in a retail setting [6]. The overall research question guiding this project is:

“Can no-code platforms like Adalo design an efficient mobile application for security guard management for SMEs in the retail industry?”

To achieve this aim, the project is guided by the following objectives:

- Analyse the current SMEs security challenges in retail environment.
- To design and develop a mobile-based guard management application using Adalo.
- To integrate other cloud solutions/services Zapier and Google Sheets for automating processes.
- To test and evaluate the usability and effectiveness of the app in improving operational efficiency and record accuracy.
- To assess the potential of no-code platforms as viable development tools for SMEs with limited technical resources.

This project will not only demonstrate how no-code tools can streamline business processes but will also contribute to the understanding of how such platforms can be

applied to security operations management, a field often underserved by technology-driven innovation. The findings are expected to provide insights into the advantages, limitations, and practical implications of no-code application development in professional environments.

CHAPTER 2 - RESEARCH AND LITERATURE REVIEW

2.1 Introduction

This chapter reviews the existing literature and scholarly work related to the development of security management systems, mobile workforce applications, and the use of no-code and low-code development platforms within computing. The purpose of this review is to situate the current project—the design and implementation of a mobile-based Guard Management App within the context of academic research and professional practice. The chapter is divided into five sections:

- Security Operations Management Systems
- Mobile Application Development in Computing
- No-Code and Low-Code Platforms
- Automation and Cloud Integration in Application Development
- Summary and Research Gaps

2.2 Security Operations Management Systems

The management of security personnel and incidents traditionally relies on manual methods such as paper-based reports and attendance registers. According to Ogunleye and Akanbi (2019), such systems are error-prone, inefficient, and difficult to maintain, leading to challenges in accountability and performance tracking [1]. The shift toward digital solutions has been driven by the need for centralization, data analytics, and real-time monitoring [2] (Adebayo et al., 2020).

Modern Guard Management Systems (GMS) automate routine security operations, such as guard patrols, clock-in/out verification, and incident logging. Hussain et al. (2021) demonstrated that cloud-based patrol systems improve situational awareness [3] and data accessibility compared to legacy systems. However, most commercial GMS platforms, such as Trackforce, Silvertrac, or GuardTek) are subscription-based and costly, making them impractical for smaller organizations [4]. This research project aims to bridge that gap by developing an affordable, customizable, and non-code-based mobile solution.

2.3 Mobile Application Development in Computing

Mobile application development has become an essential component of modern computing. Traditional app development requires programming knowledge in coding languages [5] such as Java, Kotlin, or Swift (Lee & Kim, 2018). However, the learning curve and development time are significant barriers for individuals or small organizations without software engineering expertise.

In contrast, mobile development frameworks and cloud-connected environments have simplified deployment and cross-platform compatibility. Research by Hassan and Yu (2020) highlighted that user-centered design and accessibility are key to the adoption of workplace mobile tools [6]. Similarly, Kumar et al. (2019) found that workflow-oriented mobile apps improve operational efficiency in industries such as logistics, healthcare, and security services. These findings support the use of mobile-first solutions in environments that require continuous field data collection—aligning closely with this project’s objective.

2.4 No-Code and Low-Code Platforms

The introduction of no-code and low-code platforms has transformed software engineering by enabling non-programmers to build applications through visual interfaces. Holub (2022) emphasizes that no-code tools democratize development by allowing users to rapidly design and test prototypes without programming knowledge [7]. Platforms such as Adalo, Bubble, and AppGyver have been studied for their effectiveness in small business automation [8] (Ramesh & Rajkumar, 2021). These tools utilize pre-built logic components, visual editors, and cloud-based databases to enable faster deployment.

Despite their benefits, no-code systems have limitations in customization, performance, and scalability. Carvalho and Pereira (2020) note that such tools are best suited for lightweight applications and pilot projects, particularly when the primary goal is rapid development rather than complex computation [9]. In this project, Adalo’s combination of frontend design, database management, and integration capabilities makes it an appropriate choice for prototyping and deploying a Guard Management App without traditional coding.

2.5 Automation and Cloud Integration in Application Development

Automation and cloud integration are central to modern computing practice. The use of cloud services such as Google Sheets for data storage and Zapier for workflow automation exemplifies the increasing interconnectivity of digital ecosystems. Sharma et al. (2020) demonstrated that cloud-based spreadsheets can serve as lightweight back-end databases for small-scale systems [10]. Furthermore, Zapier provides application integration via APIs, enabling non-developers to link multiple services without writing code (Zapier.com, 2025).

This form of automation supports real-time synchronization between application data and analytical tools, which is particularly valuable in guard management operations. It enables time logging, automatic calculation of work hours, and generation of performance reports, which are all critical for operational oversight. Such integrations exemplify the concept of computational thinking without code (Davis, 2021), aligning with modern computing's goal of broadening accessibility to software development [11].

2.6 Summary and Research Gaps

The literature clearly demonstrates a trend toward digital transformation and automation in operational management systems. Existing research confirms the efficiency and accessibility benefits of mobile and cloud-based solutions. However, there remains a research gap in applying no-code technologies specifically to security and guard management systems. Most studies focus on commercial solutions or traditional software engineering approaches, leaving unexplored how small organizations can leverage no-code platforms to achieve similar outcomes.

This project aims to fill that gap by demonstrating how Adalo, combined with other cloud solutions/services like Google Sheets and Zapier, can be used to develop a cost-effective, scalable, and easily deployable Security Guard Management App.

CHAPTER 3 - RESEARCH AND/OR DEVELOPMENT METHODOLOGY

3.1 Introduction

This chapter presents the research and development methodologies used in the design, implementation, and evaluation of the Guard Management Software project. The primary goal of this chapter is to describe, in sufficient detail, the methodological framework, data collection process, and analytical techniques applied throughout the study.

The project addresses recurring operational challenges in SME (Small and Medium-sized Enterprise) retail security, such as inefficient manual reporting, poor accountability, and lack of centralized communication. To achieve its objectives, this study integrates investigative research and development methodology, guided by structured observation and practical experimentation using no-code software tools.

The chosen methods ensure both academic rigor and practical relevance, enabling the project to contribute meaningfully to digital transformation within the private security sector.

3.2 Research Philosophy

The philosophical stance adopted for this project is Pragmatism, a philosophy that emphasizes the practical application of knowledge to achieve real-world outcomes. Pragmatism rejects rigid theoretical boundaries and allows researchers to use methods that best solve the identified problem [12].

In this project, the pragmatic approach aligns with the goal of developing a usable and efficient software system that directly responds to operational inefficiencies observed in SME retail security. Rather than seeking theoretical generalization, the study focuses on the usefulness, functionality, and applicability of the developed system in realistic business contexts.

Advantages:

- Encourages flexibility in combining observation, experimentation, and iterative design [13].
- Ensures the research remains outcome-oriented and solution-driven.

Disadvantages and Mitigation:

- May underemphasize theoretical depth.

Mitigation: The project maintains academic grounding by referencing established software engineering and organizational research frameworks.

3.3 Research Approach

An inductive research approach was adopted. Induction begins with specific observations and builds broader insights and conclusions [14]. This is appropriate for development-based projects, where understanding is derived from real-world practices rather than predefined hypotheses.

In this case, observations of multiple SME retail environments revealed patterns such as repetitive manual documentation, inconsistent reporting, and communication delays. These insights formed the conceptual model for the Guard Management Software, ensuring that the design was based on evidence from actual operational practice.

Advantages:

- Allows flexibility and creativity in developing solutions to observed problems.
- Supports pattern discovery in practical business contexts.

Disadvantages and Mitigation:

- Subjectivity may influence interpretation.

Mitigation: Findings were cross-referenced with academic and industry literature to validate assumptions [15], [16].

3.4 Research Strategy

The research follows an investigative strategy, focusing on exploring the challenges and inefficiencies in SME security operations. Unlike experimental or survey-based approaches, investigative research is exploratory, emphasizing understanding and interpretation rather than hypothesis testing [17].

This approach enabled the identification of key operational issues, such as paper-based inefficiencies, lack of traceability, and informal communication. These findings were then directly translated into functional system requirements.

Advantages:

- Enables comprehensive exploration of poorly documented or under-researched operational areas.
- Supports the iterative refinement of the problem definition.

Disadvantages and Mitigation:

- Findings may be context dependent.

Mitigation: The final system is designed to be modular and configurable for various retail environments.

3.5 Research Design - The Research Onion Framework

To ensure methodological coherence, the study follows the Research Onion model by Saunders et al. [18], which provides a structured, layered framework for research design.

Layer	Chosen Method	Justification
Philosophy	Pragmatism	Focuses on practical solutions to real-world security management problems.
Approach	Inductive	Builds theory and design from observation of current SME challenges.
Strategy	Investigative	Explores operational gaps and proposes a technology-driven solution.
Choice	Mixed Methods (quality observation and quantitative testing)	Combines qualitative observation with quantitative performance analysis.
Time Horizon	Cross-sectional	Research conducted within a defined project time frame
Data Collection	Structured observation and prototype performance logs	Ethically compliant and objective data sources.

This structured approach ensures logical progression from philosophical foundations to practical execution.

3.6 Data Collection Method

The primary data collection method used was structured observation. As no human subjects were interviewed or surveyed, this technique ensured full ethical compliance while allowing for meaningful operational insights. Structured observation involved systematically recording observable events in SME retail environments related to guard behavior, workflow patterns, and communication habits.

Observations focused on:

- Time spent completing manual reports.
- Frequency of missed or incomplete checklists.
- Documentation accuracy and readability.
- Communication methods between guards and supervisors.

Supplementary secondary data were obtained from relevant industry reports and literature on security operations, SME management, and digital transformation [19], [20].

Advantages:

- Ethically safe and non-intrusive.
- Provides authentic, real-world data on operational practices.

Disadvantages and Mitigation:

- Subjective interpretation may occur.

Mitigation: Structured observation templates and validation through multiple observation sessions ensured reliability.

3.7 Evaluation Criteria

The software solution was evaluated using performance-based Key Performance Indicators (KPIs) reflecting efficiency, reliability, and accuracy. These included:

1. Efficiency:

- Reduction in average task completion time after digitalization.

2. Accuracy:

- Decrease in incomplete or inconsistent reports.

3. Reliability:

- System uptime and synchronization consistency.

4. Usability:

- Ease of navigation and logical task flow during prototype testing.

3.8 Data Analysis Methods

The project employed both quantitative and qualitative data analysis to interpret results comprehensively. This mixed-methods approach allowed measurable performance assessment alongside contextual interpretation of operational patterns [21].

3.8.1 Quantitative Data Analysis

Quantitative analysis was based on metrics automatically recorded by the prototype systems. These included:

- Number of completed checklists per session.
- Average time to submit reports.
- Frequency of missed or incorrect entries.

The data were exported to Microsoft Excel, where descriptive statistical analysis was applied. Percentages were used to calculate task completion rates, while mean values

measured time efficiency improvements. Frequency counts identified recurring errors or lags in system responses.

These descriptive methods were selected as appropriate for small-scale development projects, providing clear performance trends without the need for inferential testing [22].

For example, if the average time to complete a manual report was 7 minutes, and the digital prototype reduced this to 3 minutes, this represented a 57% improvement in operational efficiency, a measurable, quantitative indicator of the system's impact.

3.8.3 Qualitative Data Analysis

The qualitative analysis followed Braun and Clarke's (2006) thematic method [23], focusing on identifying patterns within the structured observation data.

Step 1 – Raw Observations

Example observational notes:

- Guard required 10 minutes to complete paper checklist; missing pen caused delay.
- Fire door issue discussed but not logged in records.
- Supervisor unable to read unclear handwriting in logbook.
- Guards exchanged shift updates through WhatsApp messages.

Step 2 – Coding

Each observation was assigned a short code describing the issue:

Excerpt	Code	Meaning
“Checklist incomplete due to missing pen”	INE-01	Focuses on practical solutions to real-world security management problems.
“Fire door not logged”	REP-02	Missing incident reporting
“Unreadable handwriting”	DOC-04	Poor data readability
“WhatsApp used for updates”	COM-05	Combines qualitative observation with quantitative performance analysis.

Step 3 – Theme Development

Codes were grouped into broader themes:

Theme	Associated Codes	Description
Operational Inefficiency	INE-01, REP-02	Delays and omissions caused by manual reporting.

Accountability Gaps	DOC-04	Unclear and unverifiable records.
Communication Issues	COM-05	Reliance on informal messaging platforms.

Step 4 – Interpretation

Each theme was interpreted to inform system design decisions:

Theme	Observed Problem	Implemented Software Solution
Operational Inefficiency	Guards losing time on manual paperwork	Introduced digital checklist with timestamps
Accountability Gaps	Unclear paper records	Integrated digital sign-off and photo verification
Communication Issues	Informal shift handovers	Added in-app reporting and notifications

Step 5 – Integration

The thematic analysis revealed how observed inefficiencies could be transformed into actionable software requirements. By linking qualitative findings with quantitative metrics (e.g., reduced completion time and improved report accuracy), the analysis established both statistical validation and contextual understanding of system performance [18], [23].

3.9 Development Methodology

The project adopted an Agile development methodology, specifically the Rapid Application Development (RAD) model [24]. RAD emphasizes iterative prototyping, simulation feedback, and quick refinement which is ideal for no-code environments.

Rationale:

- Allows flexibility in incorporating design changes during testing.
- Encourages fast prototyping using visual development platforms like Adalo.

Advantages:

- Accelerated development cycle.
- Immediate evaluation of functionality.
- Supports continuous improvement.

Disadvantages and Mitigation:

- Documentation may lag behind rapid changes.

Mitigation: Development logs and version notes were updated after each prototype revision.

3.10 Software, Tools and Hardware

The use of no-code technologies allows non-programmers to efficiently develop fully functional applications, which aligns with the project's scope and resource constraints.

Tool	Purpose	Justification
Adalo	Mobile application builder	Simplifies checklist creation, photo uploads, and logging.
Adalo DB	Cloud database	Provides secure and scalable data storage.
Figma and Canva	Prototype and UI design	Facilitates consistent design across web and mobile platforms.
Google Sheets Microsoft Excell	Data storage	Used for data storage and descriptive analysis of test results.

No specialized hardware is required. The system is accessible via:

- Smartphones or tablets for guards (Android/iOS).

The system operates entirely online and requires only a stable internet connection to synchronize data between devices.

3.11 Validity, Reliability, and Generalizability

- **Validity:** Ensured through real-world observations and iterative prototype testing under authentic operational conditions.
- **Reliability:** Supported by consistent observation methods and reproducible test procedures.
- **Generalizability:** While focused on retail SMEs, the modular system design allows adaptation for other sectors such as logistics, hospitality, or facilities management.

3.12 Ethical Considerations

The study complied fully with institutional ethical standards. No identifiable or personal data were collected. Observations were restricted to operational behavior and workflow processes. All digital data were anonymized, and no participants were interviewed or surveyed, ensuring adherence to the principles of confidentiality, integrity, and respect for privacy [25].

3.12 Chapter Summary

This chapter detailed the methodological structure underpinning the Guard Management Software project. The integration of pragmatic philosophy, inductive reasoning, investigative research, and RAD development ensured a balanced and ethically sound approach. Quantitative system logs provided measurable evidence of improvement, while qualitative thematic analysis translated observed inefficiencies into actionable system features. The following chapter will discuss the system design and implementation, outlining the architecture, interface components, and database integration.

CHAPTER 4 – DESIGN & IMPLEMENTATION

4.1 Introduction

This chapter presents the design and implementation of the Iron Chevron Security Guard Management System, a digital solution developed to address inefficiencies in incident reporting, monitoring, and administrative oversight within a retail security environment. The chapter focuses on both the design rationale and the implementation process, demonstrating how engineering decisions were made in response to requirements, constraints, and stakeholder needs. In line with best practice in computing and systems engineering, the chapter emphasizes process as well as product, showing not only what was built, but why specific approaches were selected and how they contributed to the successful realization of the project objectives.

As this project falls within the category of development- and experimentation-based computing projects, the chapter discusses system architecture, data design, user interface development, automation workflows, and testing strategies. Rather than presenting extensive technical documentation or source code, the focus is placed on design trade-offs, integration challenges, and validation of system behavior. This approach reflects the expectations of professional computing practice, where design decisions must balance usability, scalability, maintainability, and organizational constraints.

4.2 Requirement Analysis and Design Starting Point

4.2.1 Discovery and Refinement of Requirements

The design process began with the identification of a real-world organizational problem rather than a formalized requirements document. Security operations within the retail context relied heavily on manual, paper-based processes, including handwritten incident reports, ad hoc communication, and fragmented record-keeping across departments. While these methods were familiar to staff, they introduced significant limitations, including inconsistent data quality, delays in reporting, limited visibility for management, and challenges in auditing and compliance.

Requirements were elicited through reflective analysis of current practices, supported by a review of industry literature on software engineering [26],[27], security operations, digital incident management systems, and retail loss prevention. This process enabled the transformation of a broad organizational need into a structured set of functional and non-functional requirements. Importantly, requirements were refined iteratively as system

prototypes were developed and evaluated, reflecting an agile and user-centered design philosophy.

4.2.2 Functional and Non-Functional Requirements

- Key functional requirements included:
 - Mobile-based shoplifting incident reporting
 - Mandatory data capture to ensure consistency
 - Photo evidence attachment
 - Automated timestamping and user attribution
 - Centralized administrative access
 - Role-based access control
 - Incident review, filtering, and status management
 - Exportable incident documentation
- Key non-functional requirements included:
 - Usability in operational environments
 - Reliability of data capture and synchronization
 - Security and access control
 - Scalability across multiple stores
 - Maintainability using no-code tools

These requirements directly informed subsequent design decisions, particularly in relation to system architecture and technology selection.

4.3 Architectural Design and Technology Selection

4.3.1 Architectural Overview

One of the most critical design decisions was the adoption of a distributed architecture supported by middleware automation, rather than a single monolithic application. The final architecture consists of four primary components:

- Mobile Application (Adalo) – for operational data capture by security guards
- Integration Layer (Google Sheets) – acting as a centralized data intermediary
- Automation Layer (Zapier) – responsible for data synchronization
- Web-Based Admin Application (Adalo) – for monitoring and management

This architecture was driven by both platform constraints and engineering judgement. Adalo enforces strict database isolation between applications, preventing direct sharing

of data between mobile and web apps. Rather than treating this limitation as a flaw, the project leveraged it to implement a layered architecture that mirrors real-world enterprise systems. [28]

4.3.2 Design Trade-Offs and Alternatives

Several alternative architectural approaches were considered. A single application with shared access was rejected due to poor suitability for administrative workflows. Custom backend solutions such as Firebase or Xano were also considered but dismissed due to increased development complexity, steeper learning curves, and misalignment with the project's no-code development objective.

The chosen middleware-based architecture offered several advantages:

- Clear separation of operational and administrative concerns
- Improved fault isolation
- Easier auditing and reporting
- Reduced coupling between system components
- These benefits outweighed the added complexity of managing automation workflows.

4.4 Data Design and Information Flow

4.4.1 Data Structures and Schema Design

Data design focuses on ensuring consistency, traceability, and analytical value. Each shoplifting incident record includes structured attributes such as:

- Incident date and time
- Incident narrative
- Suspect description
- Stolen items
- Estimated theft value
- Witness details
- Photo evidence
- Theft value

Care was taken to ensure that field names and data types were consistent across the mobile app, Google Sheets, and Admin web app. This consistency simplified integration and reduced the likelihood of transformation errors during synchronization.

4.4.2 Data Synchronization Process

The data flow process was deliberately designed as a one-way operational pipeline:

- Incident captured via mobile app
- Record stored in mobile database
- Automated synchronization to Google Sheets
- Replication into web app database

This approach ensures that the mobile app remains the authoritative source for incident capture, while the web app serves as a monitoring and management interface. The implemented mobile application screens are provided in Appendix-A.

4.5 System Modelling and Process Representation

4.5.1 Use Case Modelling

Use case modelling was employed to describe system behavior from a user perspective. Two primary actors were identified: Security Guard and Administrator. Each actor interacts with the system through a distinct set of use cases aligned with their responsibilities. [29]

Security guards focus on rapid data entry under operational conditions, while administrators require analytical tools and oversight capabilities. Modelling these interactions clarified system boundaries and informed screen design and navigation logic.

4.5.2 Process Modelling

Process modelling highlighted the end-to-end lifecycle of an incident, from detection to resolution. This modelling ensured that no stage of the incident management process was overlooked and that data remained consistent across systems.

4.6 User Interface and Experience Design

4.6.1 Mobile Application Design

The mobile application interface was designed with usability as the primary concern. Security officers may be operating under time pressure or in dynamic environments, requiring an interface that is intuitive and efficient. Forms were designed with:

- Vertical scrolling to accommodate long reports
- Clear labelling and grouping of fields
- Mandatory fields for critical data
- Minimal navigation depth

The decision to rely on Adalo's Form Group functionality ensured reliable record creation and reduced the risk of partial submissions.

4.6.2 Web-Based Admin Dashboard Design

The mobile application interface was designed with usability as the primary concern. Security officers may be operating under time pressure or in dynamic environments, requiring an interface that is intuitive and efficient.

The Admin Web Dashboard was designed for information visibility rather than data entry. Key features include:

- Incident lists with sorting and filtering
- Click-through detailed views
- Dashboard statistics (e.g. total theft value)
- Status indicators for incident triage

Role-based access control was enforced using conditional logic to ensure that only authorized users could access administrative functions.

4.7 Implementation Strategy and Development Process

4.7.1 Tools and Techniques

The system was implemented using a combination of no-code platforms:

- Adalo for application development
- Google Sheets for integration and reporting
- Zapier for automation workflows

An iterative development methodology was adopted, allowing features to be built incrementally and tested early. This approach reduced risk and facilitated continuous improvement. [30], [31], [32]

4.7.2 Challenges Encountered and Solutions

A significant challenge was the inability to share databases between Adalo apps. This was resolved through middleware automation. Another challenge involved ensuring reliable photo handling across systems, which was addressed by storing image URLs rather than raw files.

4.8 Data Flow Overview

The Iron Chevron Security Guard Management System uses a distributed, middleware-based data flow architecture to support operational data capture, centralized storage, and administrative oversight. (see Appendix-E)

The data flow begins at the Mobile Application, where security officers authenticate and submit structured shoplifting incident reports. These reports include mandatory textual fields, timestamps, numerical values (e.g. theft value), and optional photographic evidence. Once submitted, the data is stored locally within the mobile application's database. The underlying database structure for the system was designed to support secure, structured, and auditable data storage. The full database schema and collection definitions are provided in Appendix C.

An automation middleware layer is then used to synchronize data between systems. Zapier monitors the mobile application database and, upon detecting new incident records, transfers the data into a central Google Sheets repository. This spreadsheet acts as a neutral, persistent data store and integration hub, enabling reliable data sharing between otherwise isolated applications. (see Appendix-D)

A second automation workflow copies validated incident data from Google Sheets into the Admin Web Application database. This allows administrators to view, filter, analyze, and export incidents without accessing the mobile system or raw spreadsheets directly. Aggregated data, such as total theft value, is calculated at this stage to support management reporting.

Finally, administrators interact with the Web Admin Dashboard, where incidents can be reviewed in detail, filtered by date or store, and exported as PDF or CSV reports. This architecture ensures role separation, data consistency, and auditability while remaining scalable and maintainable.

4.9 Testing and Validation Strategy

The testing and validation strategy consisted of the following tools and techniques:

- **Unit Testing**

Individual components such as forms, buttons, and automation triggers were tested independently to verify correct behavior.

- **System Testing**

End-to-end testing validated that incidents captured in the mobile app appeared correctly in the web dashboard.

- **Usability Testing**

Usability testing focused on clarity of interface elements, form completion time, and navigation flow. Feedback informed minor layout and labelling improvements.

- **Acceptance Testing**

- Acceptance testing confirmed that the system met the original organizational need for centralized incident monitoring and reporting.

4.10 Iterative Development Overview

The project progressed through multiple iterations:

- Iteration 1: Core incident capture
- Iteration 2: Data integration and admin dashboard
- Iteration 3: Reporting and statistics
- Iteration 4: Security controls and audit logging

Each iteration improved system completeness and robustness.

4.11 Chapter Summary

This chapter demonstrates how a structured, requirements-driven design process resulted in a scalable and maintainable security incident management system. Through careful architectural planning, informed trade-offs, and iterative implementation, the project achieved its objectives while adhering to professional computing principles. The system's design and implementation provide a strong foundation for evaluation and future enhancement, which are addressed in the next chapter.

CHAPTER 5 –TESTING & EVALUATION

5.1 Introduction

This chapter presents the testing and evaluation of the Iron Chevron Security Guard Management System. The purpose of this chapter is twofold: first, to demonstrate that the system functions correctly and reliably through structured testing, and second, to critically evaluate the extent to which the implemented solution meets the project's stated aims and objectives. In line with Level 6 expectations, the chapter goes beyond a simple description of tests performed and instead analyses outcomes, compares results with findings from the literature review, and reflects on limitations, unexpected results, and opportunities for improvement.

Evaluation is conducted by comparing the system's performance and outcomes against the original project objectives and by considering how the solution aligns with existing research on digital security management systems in retail environments. This approach ensures that both technical effectiveness and academic relevance are addressed.

5.2 Testing Strategy Overview

Testing was carried out using a multi-layered strategy designed to validate system functionality, integration, reliability, and usability. Given the no-code nature of the solution, traditional code-level testing was not applicable; however, rigorous functional and scenario-based testing was conducted to ensure system correctness.

The testing strategy comprised:

- Unit testing of individual components
- System testing of integrated workflows
- Usability testing focused on end-user interaction
- Acceptance testing against defined requirements

This structured approach helped ensure that the system was complete, consistent, and fit for purposes.

5.3 Unit Testing

Unit testing focused on verifying that individual system components operated as intended when isolated from the rest of the system. This included testing:

- Mobile app form submission for shoplifting incident reports
- Mandatory field enforcement and data validation
- Photo upload functionality
- Individual Zapier automation triggers
- Web app list rendering and detail views

Each component was tested independently using test data to confirm correct behavior. For example, the shoplifting incident form was tested by submitting reports with both complete and incomplete data sets. Submissions missing required fields were correctly blocked, confirming that data validation rules were functioning as intended. [33]

Unit testing outcomes were largely positive, with issues primarily related to initial misconfiguration of form components rather than system limitations. These issues were resolved by restructuring forms using Adalo's Form Group functionality.

5.4 System Testing

System testing evaluated the behavior of the application as a whole, focusing on end-to-end workflows rather than isolated components. The primary system-level workflow tested was:

- A security guard submits a shoplifting incident via the mobile application
- The incident is stored in the mobile app database
- Zapier synchronizes the incident to Google Sheets
- A second Zap copies the data into the admin web app database
- An administrator views and manages the incident via the web interface

Multiple test incidents were submitted to validate this workflow. Results confirmed that data was transferred accurately across all stages, with no loss of information or field mismatches. Image URLs were correctly preserved, and numerical fields such as theft value were correctly aggregated in the admin dashboard statistics.

System testing demonstrated that the middleware-based architecture was reliable and robust, supporting the architectural decisions outlined in Chapter 4.

5.4 Usability Testing

Usability testing was conducted informally through scenario-based evaluation, simulating realistic usage by both security guards and administrators. The focus was on assessing whether the system was intuitive, efficient, and appropriate for its intended operational environment.

5.4 Mobile Application Usability

The mobile application was evaluated for:

- Ease of form completion
- Clarity of labels and instructions
- Navigation simplicity
- Suitability for use under time pressure

Testing showed that the scroll-based form layout was effective for long incident reports and avoided the need for multiple screens, which could disrupt operational flow. Security-related terminology was clearly understood, and mandatory fields helped ensure consistent reporting.

5.5 Web Admin Dashboard Usability

The web-based dashboard was tested for:

- Readability of incident lists
- Effectiveness of filtering and sorting
- Easy of accessing detailed incident records
- Clarity of status indicators and dashboard statistics

Administrators were able to locate and review incidents efficiently without accessing the underlying database or Google Sheets. This aligns with best practice identified in the literature, which emphasizes the importance of role-appropriate interfaces for security management systems. Administrative interfaces used for incident review are illustrated in Appendix-B.

5.6 Acceptance Testing

Acceptance testing evaluated whether the system met the original functional and non-functional requirements identified during the design phase. This test focused on answering a central question:

Does the system solve the problem it was designed to address?

Acceptance criteria included:

- Successful replacement of paper-based incident reporting
- Centralized access to incident data
- Reliable photo evidence handling
- Role-based access control
- Support for management decision-making

The system met all core acceptance criteria. In particular, the ability for administrators to view incidents, calculate total theft values, and export reports represented a significant improvement over previous manual process. [34]

5.7 Evaluation Against Objectives

Evaluation was conducted by directly comparing the system's outcomes against the project's stated objectives.

- **Objective 1: Digitize shoplifting incident reporting**
This objective was fully met. The mobile application enables structured, consistent digital reporting, eliminating reliance on handwritten records.
- **Objective 2: Centralize incident data for management oversight**
This objective was achieved through the integration of Google Sheets and the admin web app, providing a single, authoritative view of incidents across stores.
- **Objective 3: Improve data consistency and auditability**
Mandatory fields, automated timestamps, and structured data storage significantly improved data quality and traceability, supporting audit and compliance requirements.
- **Objective 4: Provide analytical and reporting capabilities**
 - Dashboard statistics and PDF export functionality enabled management-level analysis, supporting strategic decision-making.

Overall, the system successfully met all stated objectives.

5.8 Evaluation Against the Project Aim

The overarching aim of the project was to design and implement a digital security guard management solution that improves incident reporting, monitoring, and administrative control in a retail environment.

Evaluation indicates that this aim was achieved. The system not only addressed the initial operational problem but also introduced capabilities, such as automated reporting and audit trails that exceed the functionality of the original manual process.

5.9 Comparison with Literature Review Findings

Findings from this project align closely with conclusions drawn in the literature review. Prior research highlighted that digital incident management systems improve data accuracy, response times, and organizational oversight. The results of this project support these findings, demonstrating improved consistency, accessibility, and analytical capability. [35]

No significant contradictions with literature were identified. Instead, the project reinforces existing research while applying it within a new context using no-code development tools.

5.10 Unexpected Outcomes and Limitations

While the system behaved as expected, some limitations were identified:

- Dependence on third-party services (Zapier, Google Sheets)
- Synchronization delays on free automation plans
- Limited offline functionality for the mobile app

These limitations were anticipated and accepted as trade-offs given the project's constraints and objectives.

5.11 Chapter summary

This chapter has demonstrated that the Iron Chevron Security Guard Management System was rigorously tested and critically evaluated against its objectives, aims, and relevant academic literature. The results confirm that the system functions reliably, meets user needs, and provides a substantial improvement over traditional manual process. The evaluation supports the validity of the design and implementation decisions, providing a strong foundation for the concluding chapter of this dissertation.

CHAPTER 6 – CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

This chapter concludes the dissertation by drawing together the key findings, design outcomes, and evaluation results of the Iron Chevron Security Guard Management System project. It provides a reflective summary of the work undertaken, revisits the original aims and objectives, and evaluates the extent to which these were achieved. In doing so, the chapter synthesizes insights from the literature review, design and implementation, and testing and evaluation phases. Finally, the chapter identifies the benefits delivered by the system, acknowledges limitations and unresolved issues, and presents justified recommendations for future development and research.

6.2 Summary of the Project

The primary motivation for this project was the identification of inefficiencies in traditional, manual approaches to security incident reporting within a retail environment. As discussed in Chapter 1 and supported by the literature review in Chapter 2, paper-based reporting systems are prone to inconsistency, delayed communication, limited accessibility, and poor auditability. These weaknesses directly affect an organization's ability to respond to incidents, analyze trends, and make informed decisions.

In response to this problem, the project set out to design and implement a digital, centralized security guard management solution using modern no-code development tools. The resulting system consists of a mobile application for operational data capture, an administrative web application for oversight and analysis, and an integration layer that ensures reliable data synchronization between the two.

The project followed a structured development process that included requirements analysis, architectural design, system modelling, iterative implementation, and rigorous testing and evaluation. This approach ensured that the final system was not only technically functional but also aligned with organizational needs and academic best practice.

6.3 Achievement of Aims and Objectives

The overarching aim of the project was to design and implement a digital security guard management system that improves the recording, monitoring, and management of shoplifting incidents within a retail environment.

Evaluation of the completed system demonstrates that this aim was successfully achieved.

Each of the specific objectives defined in Chapter 1 was met as follows:

- **Digitization of incident reporting**
The mobile application replaced manual reporting with structured digital forms, ensuring consistent and complete data capture. Mandatory fields, automated timestamps, and photo evidence functionality addressed many of the shortcomings identified in traditional paper-based systems.
- **Centralization of incident data**
The integration of Google Sheets and the admin web application provided a single, consolidated view of incidents across stores. This significantly improved visibility and accessibility for management.
- **Improved data quality and auditability**
Structured data fields and automated synchronization improved data integrity and traceability, supporting both operational oversight and compliance requirements.
- **Provision of management and reporting tools**
Dashboard statistics, filtering capabilities, and PDF export functionality enabled management-level analysis and decision-making that was previously impractical using manual methods.

Taken together, these outcomes demonstrate that the system not only met its objectives but also delivered functionality that exceeded the capabilities of the original manual processes. [27]

6.4 Contribution to Knowledge and Practice

While the project does not claim to introduce a novel algorithm or theoretical framework, it makes a meaningful contribution in two key areas.

First, it demonstrates the practical application of no-code platforms in the development of a multi-component, role-based information system. The project shows that no-code tools, when used with appropriate architectural planning, can support complex workflows typically associated with custom-coded solutions.

Second, the project applies established concepts from security management and information systems literature to a new operational context, illustrating how digital transformation principles can be implemented in small-to-medium scale retail environments without significant technical overhead.

In this sense, the project bridges the gap between academic theory and real-world practice, reinforcing findings from the literature while extending them through applied experimentation.

6.5 Benefits of the Implemented System

The benefits delivered by the system extend beyond technical improvements and include tangible organizational advantages:

- **Operational efficiency**
Security officers can submit reports more quickly and consistently, reducing administrative burden.
- **Improved management oversight**
Administrators gain real-time access to incident data, enabling faster response and more informed decision-making.
- **Enhanced data quality**
Structured digital records reduce ambiguity and support long-term trend analysis.
- **Audit and compliance support**
Centralized records and exportable reports simplify auditing and external reporting requirements.

Importantly, these benefits are organizational rather than personal, aligning with the guidance that project outcomes should demonstrate value beyond individual skill development. [36]

6.6 Limitations and Unresolved Issues

As with any applied computing project, certain limitations remain.

One key limitation is the system's reliance on third-party platforms, such as Zapier and Google Sheets. While these tools enabled rapid development and integration, they introduce dependencies on external services, including potential limitations related to automation frequency, availability, and long-term cost.

Another limitation is the lack of offline functionality within the mobile application. In environments with poor connectivity, incident submission may be delayed until network access is restored.

Additionally, while usability testing indicated positive outcomes, formal large-scale user testing was beyond the scope of this project due to time and access constraints.

These limitations do not undermine the validity of the project but highlight realistic constraints commonly encountered in real-world system development.

6.7 Recommendations and Suggestions for Further Work

Although the Iron Chevron Security Guard Management System successfully meets its stated aims and objectives, the evaluation process identified several opportunities for further development and research. These recommendations reflect both technical enhancements and broader system improvements that could be pursued if additional time, resources, or organizational support were available. Importantly, these suggestions are grounded in observations made during system testing and evaluation and are consistent with trends identified in academic and professional literature.

6.7.1 Technical Enhancements

- **Integration of a dedicated backend service**

One significant area for future development is the integration of a dedicated backend service to replace or complement the current middleware-based architecture. While the use of Google Sheets and Zapier provided a practical and accessible solution, a purpose-built backend (such as Firebase or Xano) could enable real-time data synchronization, reduce dependency on third-party automation intervals, and improve overall system performance. Such an enhancement would be particularly beneficial in environments where immediate incident visibility is critical for operational decision-making. [37]

- **Offline data capture**

Another important enhancement involves the implementation of offline data capture with deferred submission. At present, the mobile application requires an active network connection to submit incident reports. Introducing offline functionality would allow security officers to record incidents in environments with limited connectivity, with data automatically synchronized once a connection is restored. This capability would improve system reliability and usability in real-world operational contexts.

- **Advanced analytics and reporting features**

Further development could also include advanced analytics and reporting features, such as trend analysis, heat maps of incident locations, and predictive loss modelling. By analyzing historical incident data, the system could support proactive security planning and resource allocation. Such analytical capabilities align with

content management on data-driven security management and could significantly enhance the strategic value of the system.

- **Enhanced role-based permissions**

Finally, the system could be strengthened through enhanced role-based permissions and audit controls. While basic access control is currently implemented, future iterations could introduce granular permissions, multi-level approval workflows, and more detailed audit logs. These features would improve accountability, support compliance with organizational policies, and make the system more suitable for larger or more regulated retail environments.

6.8 Chapter summary

This chapter concluded the dissertation by summarizing the project, evaluating the achievement of aims and objectives, identifying benefits and limitations, and presenting recommendations for future work. The Iron Chevron Security Guard Management System demonstrates the effective application of computing principles to solve a real-world problem, reinforcing the relevance of applied computing within modern organizational contexts.

CHAPTER 7 – REFLECTION

7.1 Introduction

This chapter provides a critical reflection on the development of the Iron Chevron Security Guard Management System and the learning gained throughout the project lifecycle. Reflection is an essential component of applied computing projects, as it enables the evaluation of both technical outcomes and personal development. Rather than focusing solely on what was produced, this chapter examines how the project experience contributed to new skills, changes in perspective, and a deeper understanding of the relationship between academic theory and professional practice. It also considers how the project could have been improved and how the learning gained will inform future work.

7.2 Reflection on Learning and Skill Development

One of the most significant outcomes of this project was the development of a broader understanding of systems thinking within computing. At the outset, the project was perceived primarily as a technical task involving application development. However, as the project progressed, it became clear that successful system development depends equally on requirements analysis, architectural planning, stakeholder needs, and iterative evaluation. [38]

From a technical perspective, the project significantly enhanced skills in no-code application development, particularly through the use of Adalo. Prior to the project, understanding of no-code platforms was largely conceptual. Through hands-on implementation, this understanding evolved into practical competence, including database design, form handling, conditional logic, and user interface structuring. The project demonstrated that no-code development does not eliminate the need for engineering discipline; rather, it shifts the focus toward architectural reasoning and process design. [39]

In addition, the integration of automation and middleware tools, such as Google Sheets and Zapier, contributed to a deeper appreciation of modern distributed systems. Designing reliable data flows across multiple platforms requires careful consideration of data consistency, error handling, and synchronization timing. This experience reinforced theoretical concepts from systems integration and enterprise architecture discussed in the literature review.

7.3 Linking Theory to Practice

A key learning outcome was the ability to connect academic theory with real-world implementation. Concepts identified in the literature review, such as the limitations of manual reporting systems, the importance of centralized data management, and the role of digital tools in improving organizational efficiency were directly observed during the design and evaluation stages of the project.

For example, academic research highlighted that manual security reporting often leads to inconsistent data and limited analytical value. This was reflected in the initial problem context and subsequently addressed through structured digital forms and mandatory data fields. Similarly, literature emphasizing role-based access control and auditability informed the separation between mobile operational users and administrative users within the system.

This alignment between theory and practice strengthened confidence in the validity of the design decisions and reinforced the importance of evidence-based system development.

7.4 Reflection on Design and Decision-Making

The design phase was particularly instructive in highlighting the importance of trade-off analysis. Many design decisions involved balancing ideal solutions against practical constraints such as platform limitations, time availability, and technical skill levels. For example, while a custom-coded backend could have offered greater flexibility, the decision to adopt a no-code, middleware-based architecture was justified by the project's scope and objectives.

Initially, platform limitations, such as Adalo's isolated databases, were perceived as obstacles. However, reflecting on this experience revealed that such constraints encouraged more thoughtful architectural design. The use of an integration layer mirrored real-world enterprise systems and introduced concepts of decoupling and modularity that may not have been explored in a simpler, single-application solution.

This shift in perspective represents a key learning point: effective engineering is not about eliminating constraints, but about designing robust solutions within them.

7.5 Reflection on Problem-Solving and Adaptability

Throughout the project, numerous challenges were encountered, particularly during implementation and integration. These included issues with form submission, data synchronization delays, and access control logic. Initially, such problems were frustrating and time-consuming. However, reflecting on these experiences highlights the development of problem-solving resilience and adaptability.

Rather than abandoning approaches when issues arose, problems were addressed through systematic troubleshooting, consultation of documentation, and iterative testing. This process improved confidence in diagnosing issues and reinforced the value of persistence and structured debugging approaches.

In retrospect, earlier prototyping and more frequent testing cycles could have reduced some of these difficulties. This insight will inform future projects by encouraging earlier validation of core workflows before expanding system functionality.

7.6 Personal and Professional Development

Beyond technical skills, the project contributed to significant personal and professional growth. The process of independent planning, executing, and documenting a substantial project enhanced time management, self-discipline, and critical thinking. The need to justify design decisions academically also improved written communication and the ability to articulate complex ideas clearly.

The project also changed perceptions of what constitutes professional software development. Rather than viewing coding ability as the sole measure of competence, the project demonstrated the importance of analysis, design, evaluation, and communications skills that are equally valued in industry settings.

This shift in mindset represents an important developmental outcome and aligns closely with the expectations of graduate-level computing professionals. [39]

7.7 Reflection on What Could Have Been Done Better

While the project was successful, reflection identifies several areas where it could have been improved. More extensive usability testing involving real security personnel would have provided richer feedback and strengthened the evaluation. Additionally, earlier consideration of reporting and export requirements could have reduced the need for later architectural adjustments.

Time constraints also limited exploration of advanced features such as predictive analytics or offline functionality. In future projects, clearer prioritization and scope management could help ensure that core enhancements are addressed earlier in the development cycle.

Acknowledging these limitations demonstrates an understanding that no project is entirely complete and that reflective evaluation is essential for continuous improvement.

7.8 Learning for Future Projects

If the project were to be undertaken again, several changes would be implemented. These include earlier requirements validation with stakeholders, increased use of low-fidelity prototypes before fully implementation, and more structured evaluation planning. Additionally, future work would benefit from exploring hybrid development approaches that combine no-code platforms with lightweight custom backend services.

Most importantly, the project reinforced the value of reflective practice itself. By critically examining both successes and shortcomings, it becomes possible to approach future projects with greater confidence, realism, and technical maturity.

7.9 Chapter summary

This chapter reflects on the learning gained throughout the project, highlighting both technical and personal development. The project deepened understanding of applied computing principles, strengthened problem-solving abilities, and fostered a more professional approach to system design and evaluation. Through reflection, it is evident that the project not only achieved its technical objectives but also contributed meaningfully to long-term academic and professional growth.

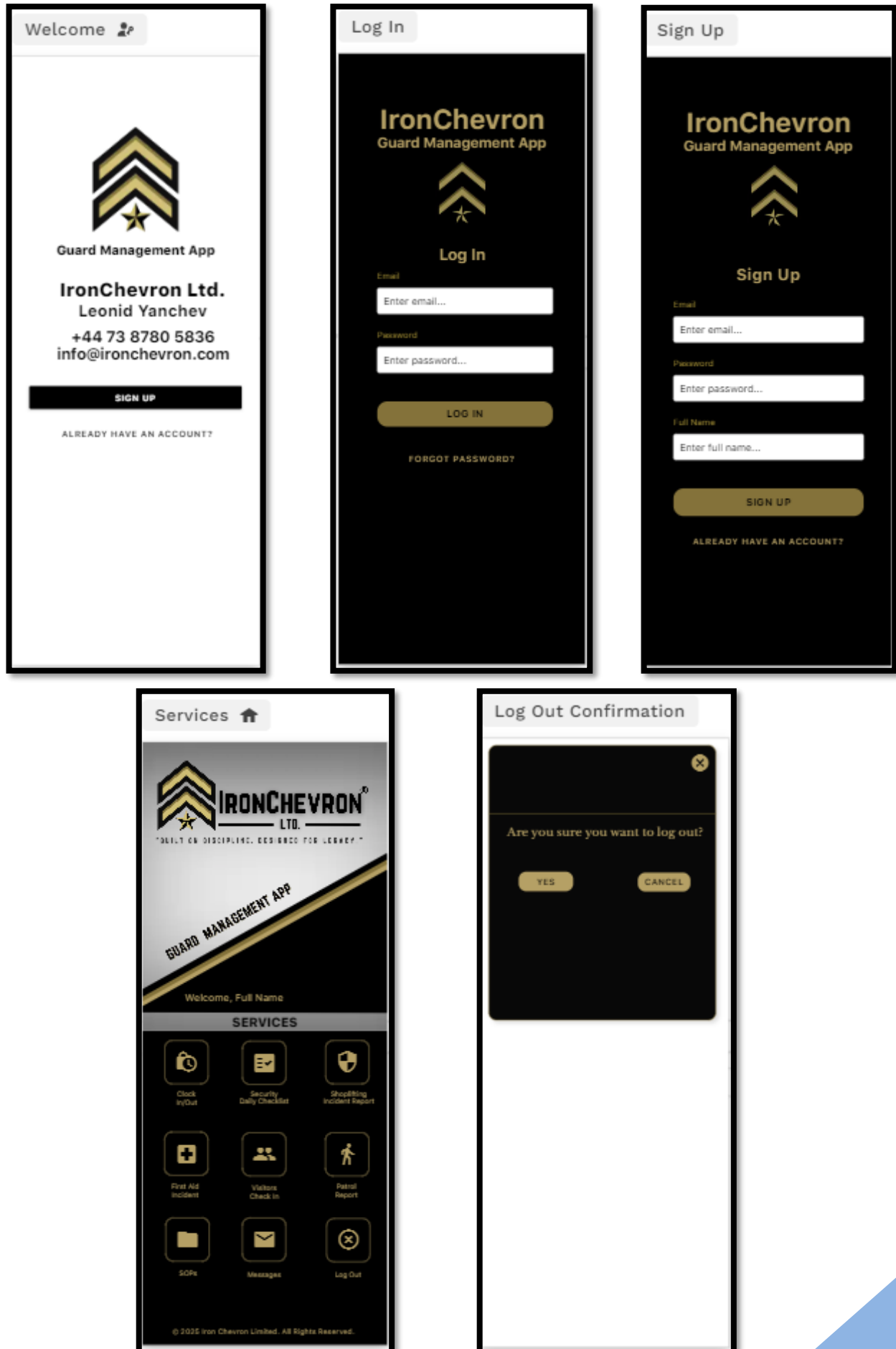
REFERENCES

- [1] O. Ogunleye and T. Akanbi, “Design and implementation of a digital patrol management system,” *Journal of Security Technologies*, vol. 6, no. 2, pp. 45–52, 2019.
- [2] R. Adebayo, A. Musa, and O. Balogun, “Digitizing security operations for real-time oversight,” *Computing and Information Systems Review*, vol. 12, no. 4, pp. 110–118, 2020.
- [3] A. Holub, “The future of software development: No-code and low-code platforms,” *IEEE Software*, vol. 39, no. 2, pp. 84–87, 2022.
- [4] Adalo Inc., “Adalo for Startup Founders,” [Online]. Available: <https://www.adalo.com/startup-founders> . [Accessed: 04-Aug-2025].
- [5] Make.com, “Automation Platform for Business Workflows,” *Make*, 2025. [Online]. Available: <https://www.make.com/en>. [Accessed: 04-Aug-2025].
- [6] M. Ramesh and R. Rajkumar, “A study on the efficiency of no-code platforms for rapid application development,” in *Proc. 6th Int. Conf. Inventive Computation Technologies (ICICT)*, 2021, pp. 1352–1357.
- [7] J. Lee and H. Kim, “Cross-platform mobile application development: A survey,” *IEEE Software*, vol. 35, no. 3, pp. 14–23, 2018.
- [8] A. Hussain, P. Roberts, and L. Zhang, “Cloud-enabled guard patrol and reporting systems,” *IEEE Access*, vol. 9, pp. 14432–14441, 2021.
- [9] L. Carvalho and P. Pereira, “Evaluating scalability in no-code applications,” *International Journal of Emerging Computer Systems*, vol. 4, no. 1, pp. 33–41, 2020.
- [10] A. Sharma, P. Kumar, and R. Kumar, “Google Sheets as a lightweight backend for prototyping web apps,” in *Proc. 11th Int. Conf. Comput., Commun. and Netw. Technol. (ICCCNT)*, 2020, pp. 1–5.
- [11] S. Davis, *Computational Thinking in No-Code Environments*, New York: Springer, 2021.

- [12] W. James, *Pragmatism: A New Name for Some Old Ways of Thinking*. New York: Longmans, Green & Co., 1907.
- [13] J. Dewey, *The Quest for Certainty*. New York: Minton, Balch & Co., 1929.
- [14] J. W. Creswell and J. D. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 5th ed. Thousand Oaks, CA: Sage, 2018.
- [15] British Security Industry Association (BSIA), *Best Practice Guidelines for Security Guarding Services*, 2022.
- [16] SIA (Security Industry Authority), *Private Security Operational Standards*, London, UK, 2021.
- [17] D. Wilson, *Investigative Research: Understanding Inquiry through Exploration*. London: Routledge, 2016.
- [18] M. Saunders, P. Lewis, and A. Thornhill, *Research Methods for Business Students*, 9th ed. Harlow, UK: Pearson Education, 2019.
- [19] P. Drucker, *Innovation and Entrepreneurship*. New York: HarperCollins, 2015.
- [20] D. Storey, *Understanding the Small Business Sector*. London: Routledge, 2016.
- [21] J. W. Creswell and V. L. Plano Clark, *Designing and Conducting Mixed Methods Research*, 3rd ed. Thousand Oaks, CA: Sage, 2018.
- [22] A. Field, *Discovering Statistics Using IBM SPSS Statistics*, 5th ed. London: Sage, 2018.
- [23] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006.
- [24] J. Martin, *Rapid Application Development*. New York: Macmillan, 1991.
- [25] B. G. Buchanan and M. Zimmer, "Internet research ethics," in *Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., 2021.
- [26] I. Sommerville, *Software Engineering*, 10th ed. Harlow, UK: Pearson Education, 2016.

- [27] R. Pressman and B. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York, NY, USA: McGraw-Hill, 2020.
- [28] M. Fowler, *Patterns of Enterprise Application Architecture*. Boston, MA, USA: Addison-Wesley, 2003.
- [29] A. Cockburn, *Writing Effective Use Cases*. Boston, MA, USA: Addison-Wesley, 2001.
- [30] Adalo, "No-Code App Development Platform," 2025. [Online]. Available: <https://www.adalo.com> [Accessed: 04-Aug-2025].
- [31] Google, "Google Sheets: Cloud-Based Spreadsheet Application," 2025. [Online]. Available: <https://www.google.com/sheets> [Accessed: 04-Aug-2025].
- [32] Zapier, "Workflow Automation Platform," 2025. [Online]. Available: <https://zapier.com> [Accessed: 04-Aug-2025].
- [33] G. J. Myers, C. Sandler, and T. Badgett, *The Art of Software Testing*, 3rd ed. Hoboken, NJ, USA: Wiley, 2011.
- [34] D. Avison and G. Fitzgerald, *Information Systems Development: Methodologies, Techniques and Tools*, 4th ed. Maidenhead, UK: McGraw-Hill, 2006.
- [35] A. Dehghani and M. Ramsin, "Evaluation of Software Testing Approaches in Agile and Traditional Development," *Journal of Systems and Software*, vol. 85, no. 1, pp. 222–238, 2012.
- [36] M. Porter, *Competitive Advantage: Creating and Sustaining Superior Performance*. New York, NY, USA: Free Press, 2008.
- [37] ISO/IEC, "ISO/IEC 25010: Systems and Software Quality Models," Geneva, Switzerland, 2011
- [38] I. Sommerville, *Software Engineering*, 10th ed. Harlow, UK: Pearson Education, 2016.
- [39] A. Kolb, *Experiential Learning: Experience as the Source of Learning and Development*, 2nd ed. Upper Saddle River, NJ, USA: Pearson Education, 2015.

Appendix A – Mobile Application Screens (Adalo)



Appendix-A Mobile Application Screens (Adalo)-cont.

Clock In Out



CLOCK IN/OUT

Current time is: Current Time



CLOCK IN



CLOCK OUT

Total Hours for This Month:

BACK TO SERVICES

LOG OUT

Shoplifting Incident Report



SHOPLIFTING INCIDENT REPORT

Incident Title

Enter incident title...

Date Time

Thursday, May 4th, 2024 at 4:00 PM

Location of the theft

Enter location of the theft...

Incident Description

Enter incident description...

Suspect Description

Enter suspect description...

Stolen Items

Enter stolen items...

Photo Evidence

Choose Photo

Witness Details

Enter witness details...

Theft Value

Enter theft value...

CREATE SHOPLIFTING INCIDENT REPORT

Appendix B – Admin Web Dashboard Screens (Adalo)

 Admin Sign In  



IRONCHEVRON[®]
LTD.
"BUILT ON KIDSLIFE. DESIGNED FOR KIDZ."

Sign Up

Email

Password

Full Name

SIGN UP

ALREADY HAVE AN ACCOUNT?

 Admin Log In



IRONCHEVRON[®]
LTD.
"BUILT ON KIDSLIFE. DESIGNED FOR KIDZ."

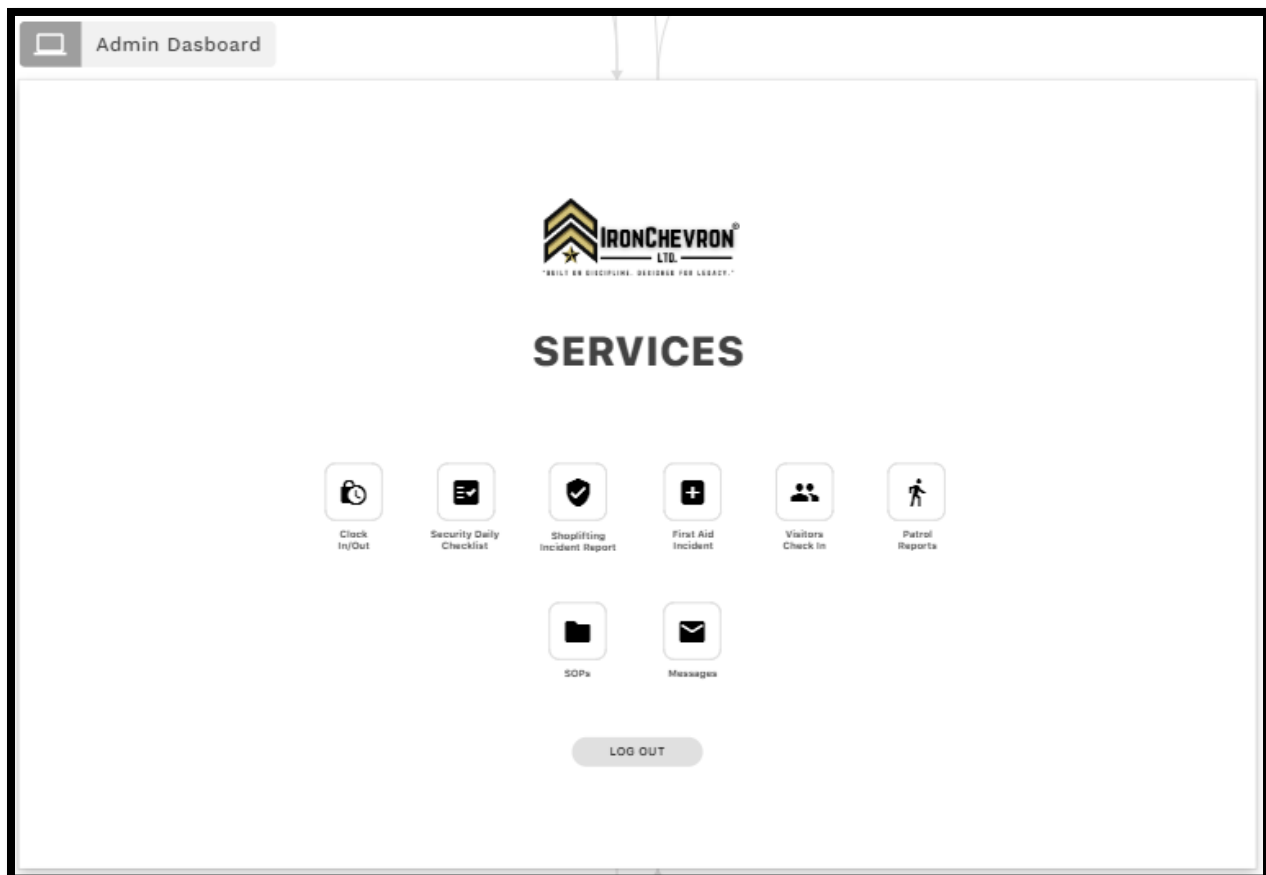
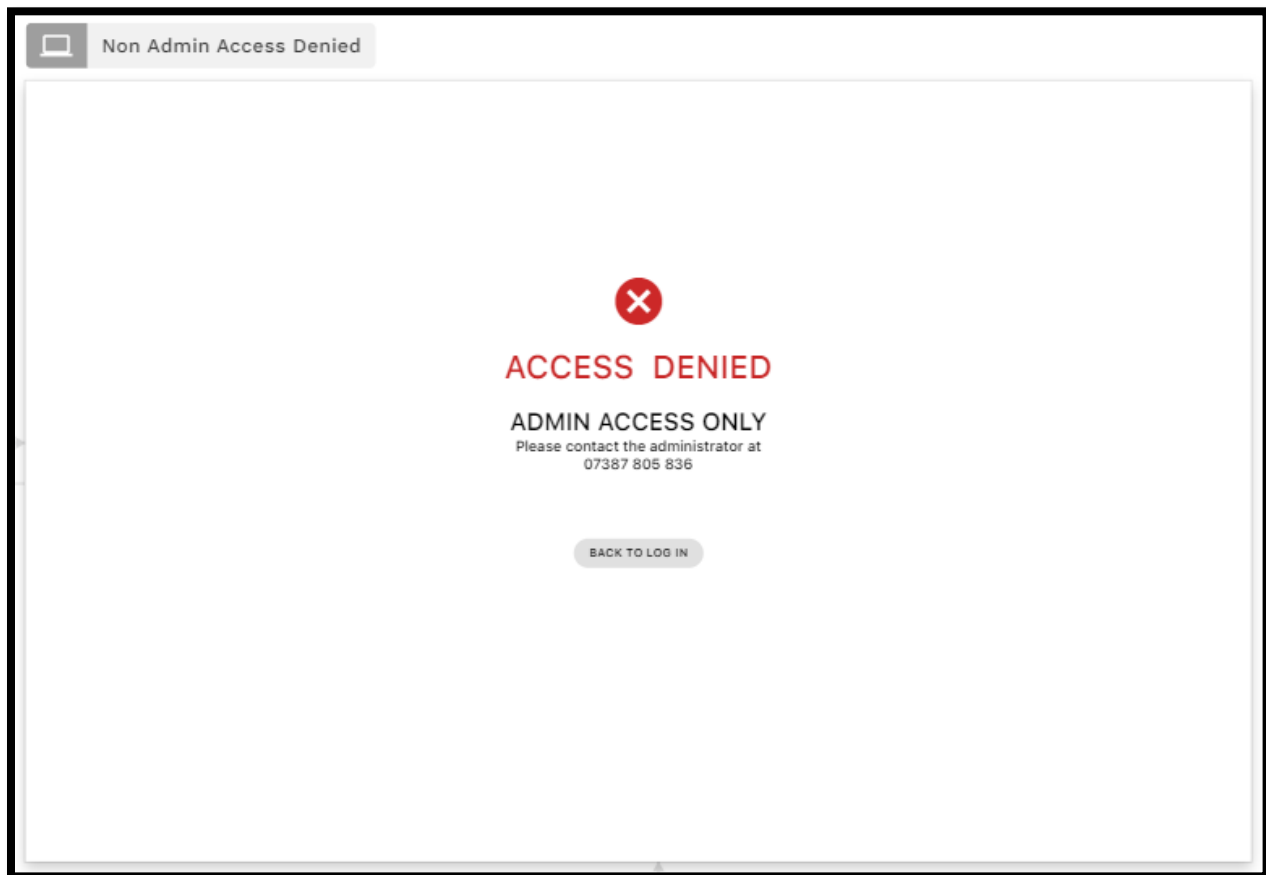
Log In

Email

Password

LOG IN

Appendix-B Admin Web Dashboard Screens (Adalo)-cont.



Appendix-B Admin Web Dashboard Screens (Adalo) cont.

Shoplifting Incidents

SHOPLIFTING INCIDENTS

Date Time	Location of the theft	Incident Description	Suspect Description	Stolen Items	Photo Evidence	Witness Details	Theft Value ↓
1 day ago	Menswear Department	Female in her 50s entered the store at 09:45 via LGF and concealed a Tommy Hilfiger men's jacket.	Female in her 50s with long blonde hair, green tracksuit, and white trainers.	Tommy Hilfiger jacket		Anastasia Anastasova	£220.00
1 week ago	Menswear Department	A group of three individuals entered via the LGF entrance, and one suspect concealed one men's shirt set under his jacket.	Male in his late 40s wearing blue jeans and a green bomber jacket.	Hugo Boss men's shirt		Samuel Samoilovski	£155.00
7 hours ago	Home Department	Female in her 40s entered via the LGF entrance, approached the display, and concealed one hair trimmer in her bag.	Female in her 40s wearing blue jeans, a black hooded jacket, and navy boots	Hair trimmer		Tino Tinovski	£120.00
2 weeks ago	Home Department	Male in his 20s entered the store via the LGF entrance and concealed one package of a wireless microphone in his jacket.	Male in his 20s wearing a black Adidas tracksuit and white trainers. Stolen Items	Wireless Microphone		Lily Lilovska	£45.00
2 weeks ago	Cosmetics Department	Female in her 50s entered the store at 18:30 via LGF, went to the Cosmetics section via the escalators, and concealed 1 x Nivea lotion.	Female in her 50s with long blonde hair, green tracksuit, and white trainers	Nivea Lotion		Amber Amberovskaya	£45.00

BACK TO SERVICES

Appendix C – Database Design & Collections (Adalo)

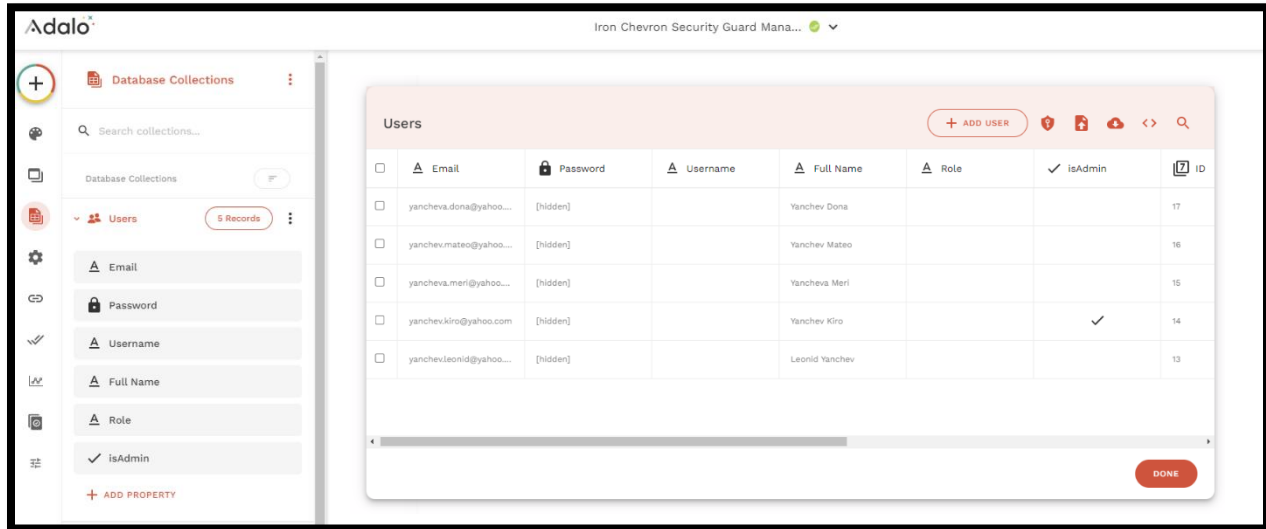


Figure C-1 Users database design in Adalo

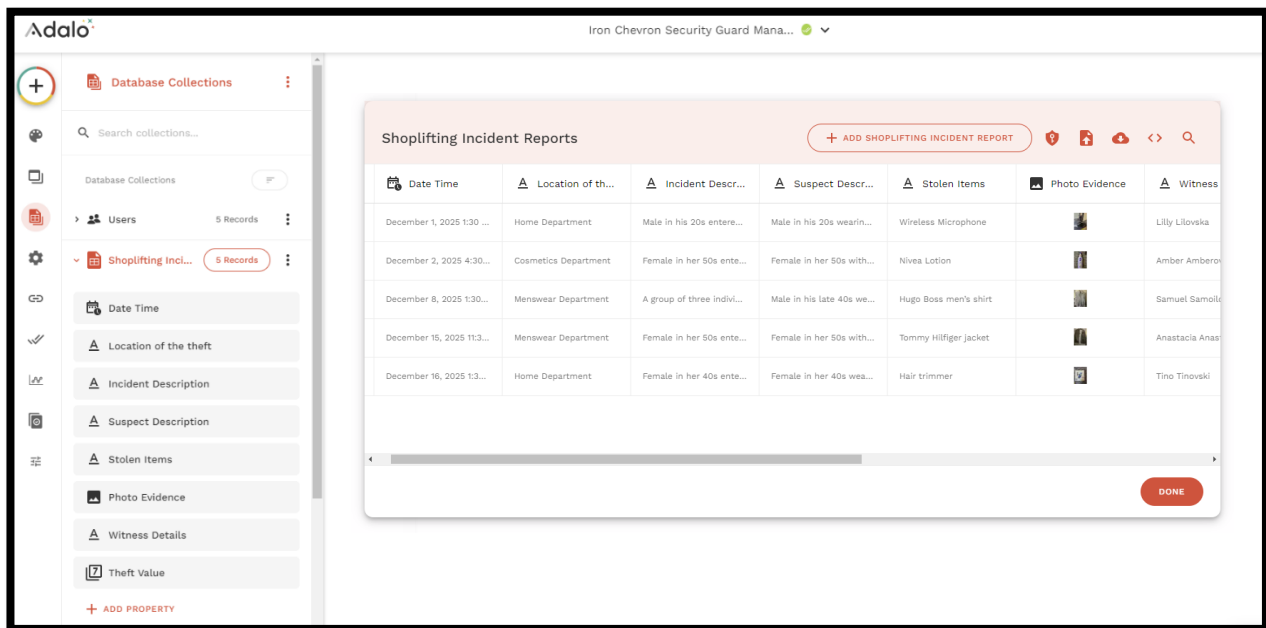
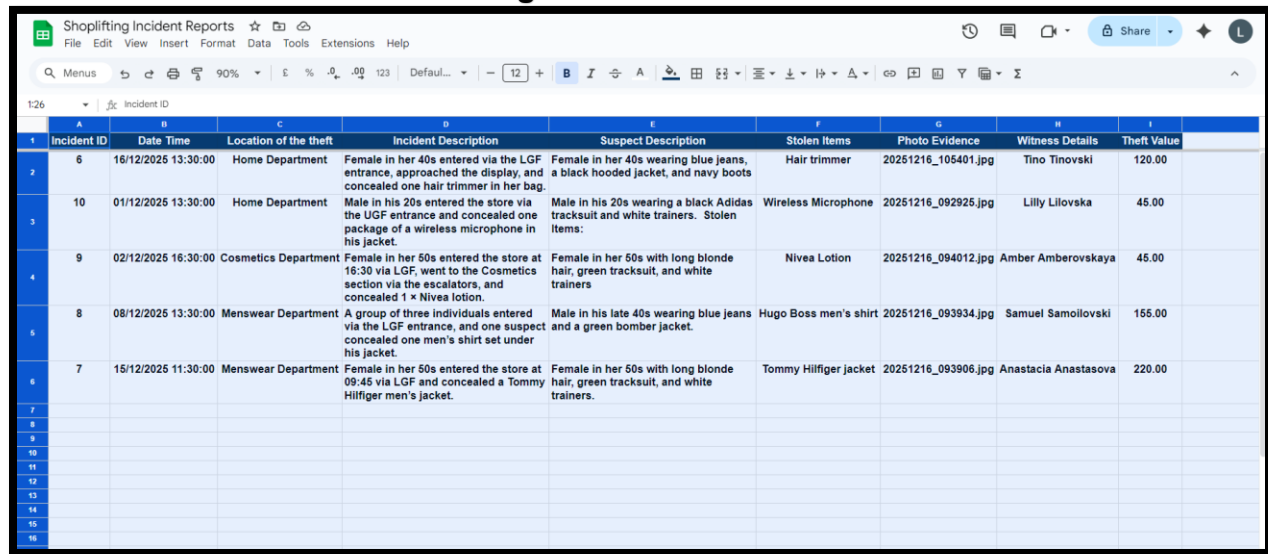


Figure C-2 Shoplifting Incident Reports database design in Adalo

Appendix-D – Integration & Automation Evidence

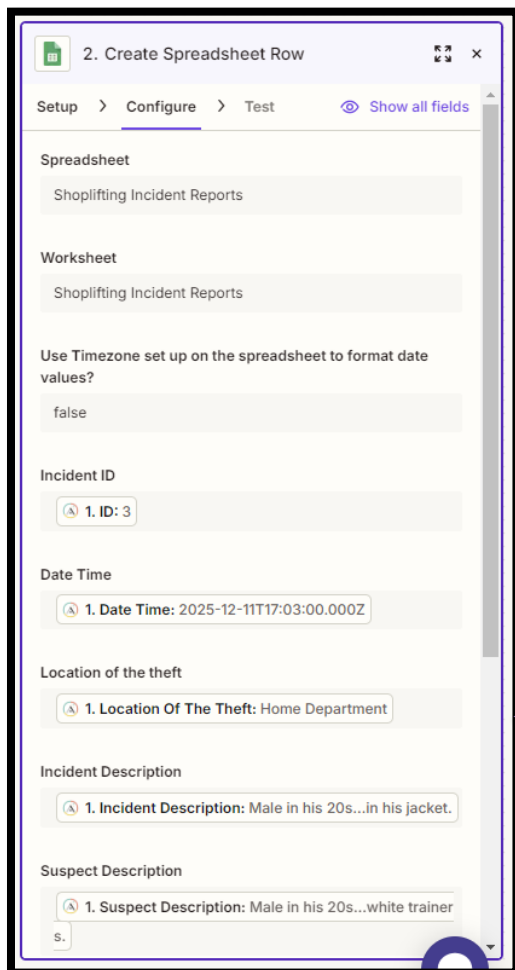
Google Sheets structure



Incident ID	Date Time	Location of the theft	Incident Description	Suspect Description	Stolen Items	Photo Evidence	Witness Details	Theft Value
6	16/12/2025 13:30:00	Home Department	Female in her 40s entered via the LGF entrance, approached the display, and concealed one hair trimmer in her bag.	Female in her 40s wearing blue jeans, a black hooded jacket, and navy boots	Hair trimmer	20251216_105401.jpg	Tino Tinovski	120.00
10	01/12/2025 13:30:00	Home Department	Male in his 20s entered the store via the UGF entrance and concealed one package of a wireless microphone in his jacket.	Male in his 20s wearing a black Adidas tracksuit and white trainers. Stolen items:	Wireless Microphone	20251216_092925.jpg	Lilly Lilovska	45.00
9	02/12/2025 16:30:00	Cosmetics Department	Female in her 50s entered the store at 16:30 via LGF, went to the Cosmetics section via the escalators, and concealed 1 x Nivea lotion.	Female in her 50s with long blonde hair, green tracksuit, and white trainers	Nivea Lotion	20251216_094012.jpg	Amber Ambersovskaya	45.00
8	08/12/2025 13:30:00	Menswear Department	A group of three individuals entered via the LGF entrance, and one suspect concealed one men's shirt set under his jacket.	Male in his late 40s wearing blue jeans and a green bomber jacket.	Hugo Boss men's shirt	20251216_093934.jpg	Samuel Samoilovski	155.00
7	15/12/2025 11:30:00	Menswear Department	Female in her 50s entered the store at 09:45 via LGF and concealed a Tommy Hilfiger men's jacket.	Female in her 50s with long blonde hair, green tracksuit, and white trainers.	Tommy Hilfiger Jacket	20251216_093906.jpg	Anastacia Anastasova	220.00

Figure D-1 Shoplifting Incident Reports triggered in Google Sheets via Zapier

Zapier workflow automation



2. Create Spreadsheet Row

Setup > Configure > Test

Spreadsheet

Shoplifting Incident Reports

Worksheet

Shoplifting Incident Reports

Use Timezone set up on the spreadsheet to format date values?

false

Incident ID

1. ID: 3

Date Time

1. Date Time: 2025-12-11T17:03:00.000Z

Location of the theft

1. Location Of The Theft: Home Department

Incident Description

1. Incident Description: Male in his 20s...in his jacket.

Suspect Description

1. Suspect Description: Male in his 20s...white trainer

Figure D-2 Configuring Google sheets record in Zapier



1. New Record

Setup > Configure > Test

App

Iron Chevron Security Guard Management

Collection

Shoplifting Incident Reports

Figure D-3 Configuring Adalo records in Zapier

Appendix E – Data Flow Overview Diagram (Logical Representation)

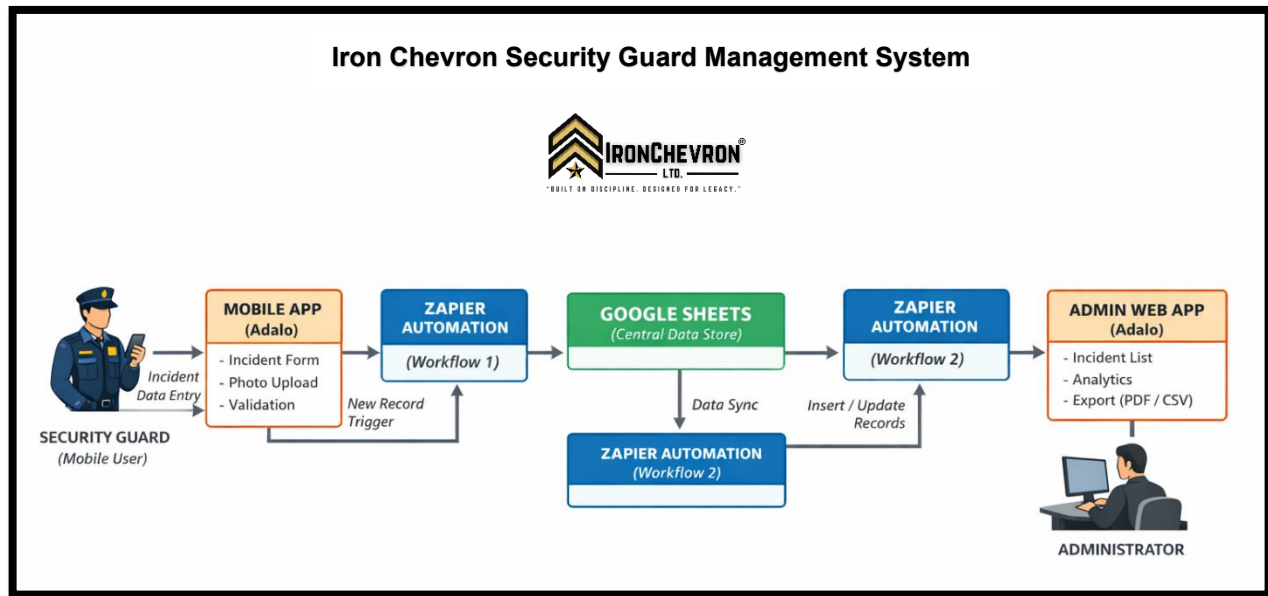


Figure E-1 Data Flow Overview Diagram

Appendix F – Project Timeline (GANTT Chart)

