



“Good morning, all, my name is Leonid Yanchev, level 6 student in Applied Computing Program at the UWTSD-London, and today I’ll be presenting the progress of my project — the design and development of a Security Guard Management Application intended to improve security operations in SMEs in the retail industry.”

1. PROJECT OVERVIEW

- ✓ **Addresses inefficiencies in manual guard reporting**
- ✓ **Transforms paper-based logs into a digital cloud system**
- ✓ **Enables real-time reporting, traceability, and accountability**

“This project focuses on developing a digital tool that replaces traditional paper-based security reporting with a cloud-based, automated solution. It allows security guards to log their activities such as security daily reports, security incidents, patrol reports etc. in real time, upload photos for verification, and enables supervisors to monitor performance remotely”

2. RATIONALE FOR CHOOSING THE PROJECT

- ✓ **Based on real-world experience in security management**
- ✓ **Observed inefficiencies in manual SME reporting**
- ✓ **Combines my background in security with computing studies**

“I chose this topic because of my own professional background in security operations, where I’ve observed that many small businesses still rely on outdated manual reporting systems. I wanted to create a practical solution that directly improves their day-to-day workflow and accountability regarding security operations”

3. PROJECT AIM

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.

My 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?”

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 . 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?”

4. PROJECT OBJECTIVES

- 1. Analyze the current SME security challenges in retail environment.**
- 2. Design and develop a mobile-based guard management application using Adalo.**
- 3. To integrate other cloud solutions/services for automating processes.**
- 4. To test and evaluate the usability and effectiveness of the app in improving operational efficiency and record accuracy.**
- 5. To assess the potential of no-code platforms as viable development tools for SMEs with limited technical resources.**

To achieve this aim of this project I was guided by the following objectives:

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

5. WHO WILL BENEFIT FROM THE PROJECT

- ✓ **SME retail managers – improved oversight.**
- ✓ **Security guards – simplified reporting.**
- ✓ **Clients – increased transparency.**
- ✓ **Security companies – better compliance documentation.**

“The beneficiaries extend beyond just guards — managers gain clearer insight, clients receive verified reports, and the system supports compliance with industry standards.”

6. RESEARCH AND LITERATURE REVIEW

Key Literature insights:

Source	Contribution
British Security Industry Association, <i>Security Guarding Guidelines</i> , 2022	Security reporting best practices
Security Industry Authority, <i>Operational Standards</i> , 2021.	Accountability and traceability standards
A. Holub, "The future of software development: No-code and low-code platforms," <i>IEEE Software</i> , vol. 39, no. 2	No-code development trends
M. Ramesh and R. Rajkumar, "A study on the efficiency of no-code platforms for rapid application development," in <i>Proc. 6th Int. Conf. Inventive Computation Technologies (ICICT)</i> , 2021	No-code platforms application development
L. Carvalho and P. Pereira, "Evaluating scalability in no-code applications," <i>International Journal of Emerging Computer Systems</i> , vol. 4, no. 1	Limitations of no-code systems in customization, performance, and scalability.
A. Sharma, P. Kumar, and R. Kumar, "Google Sheets as a lightweight backend for prototyping web apps," in <i>Proc. 11th Int. Conf. Comput., Commun. and Netw. Technol. (ICCCNT)</i> , 2020	Use of cloud solutions/services for data storage and workflow automation while increasing interconnectivity of digital ecosystems

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 can be used to develop a cost-effective, scalable, and easily deployable Security Guard Management Application.

6.1 COMPARISON OF TRADITIONAL MANUAL METHODS VS MODERN DIGITAL GUARD MANAGEMENT SYSTEMS

Criteria	Traditional Manual Methods	Modern Digital Management Systems
Data Recording	Relies on handwritten logs, attendance sheets, and paper reports	Uses digital forms, databases, and automated time-stamping
Accuracy	Prone to human error, missing entries, and falsified records.	High accuracy through automation, validation checks, and time-tracking
Efficiency	Time-consuming data entry and report compilation.	Real-time updates, instant data access, and automatic calculations
Data Storage & Retrieval	Physical storage (files, binders) prone to loss or damage; difficult to retrieve historical data.	Cloud storage ensures secure, centralized access and long-term archiving.
Monitoring & Oversight	Supervisors rely on manual inspection or physical presence.	Managers can remotely monitor activity and generate analytics reports.
Security & Privacy	Paper files can be lost, stolen, or viewed by unauthorized persons.	Data encryption and user authentication enhance security and compliance.

This comparison demonstrates that while **traditional manual methods** are simple and familiar, they lack the accuracy, scalability, and efficiency required for modern multi-site operations. In contrast, **digital guard management systems**—particularly those built using no-code platforms like **Adalo** offer real-time data synchronization, automation, and centralized oversight. These advantages make them more suitable for organizations seeking cost-effective and scalable security management solutions.

The digital approach also supports data-driven decision-making, aligning with contemporary computing practices in automation and information systems.

6.2 COMPARISON OF CODED VS. NO-CODE SECURITY GUARD MANAGEMENT PLATFORMS

Criteria	Coded Platforms (Traditional Software Development)	No-Code Platforms (e.g., Adalo, Bubble, Glide)
Development Process	Built using programming languages (e.g., Java, Python, Swift) and frameworks; requires professional developers	Created visually using drag-and-drop interfaces without coding knowledge.
Development Time	Long — may take several months to design, code, test, and deploy.	Rapid — prototypes and apps can be built within days or weeks.
Cost	High due to developer salaries, software licenses, and maintenance fees.	Low to moderate — subscription-based pricing; minimal human resource cost
Customization	Fully customizable; developers can modify the codebase for unique features.	Limited to platform capabilities; advanced custom features may be restricted.
Scalability	Highly scalable; can support large databases and enterprise-level use.	Moderately scalable; suitable for small to medium organizations.

Traditional **coded platforms** provide greater flexibility, scalability, and performance for complex enterprise-grade solutions. However, they demand significant technical expertise, financial resources, and time for development and maintenance.

In contrast, **no-code platforms** like **Adalo** democratize software creation by allowing non-technical users to design and deploy applications rapidly. They are ideal for **small to medium-sized organizations** seeking affordable, easy-to-maintain solutions without investing in full development teams. For the context of this project, the **no-code approach** aligns with Iron Chevron Limited operational needs—offering a cost-effective, scalable solution to improve daily security management without relying on extensive IT infrastructure.

6.2 COMPARISON OF CODED VS. NO-CODE SECURITY GUARD MANAGEMENT PLATFORMS-cont.

Criteria	Coded Platforms (Trackforce, Silvertrac, or GuardTek)	No-Code Platforms (e.g., Adalo, Bubble, Glide)
Integration Capabilities	Can integrate with any third-party APIs or databases through custom code.	Integrations are limited to pre-built connectors (e.g., Google Sheets, Make.com, Zapier).
Performance	Optimized for speed and heavy processing; dependent on code efficiency.	Generally slower than coded apps; dependent on the no-code platform's infrastructure.
Security	Can implement strong, custom security protocols and encryption.	Relies on built-in platform security measures and compliance standards.
Maintenance	Requires ongoing developer involvement for bug fixes and updates.	Low maintenance — updates and hosting handled by the no-code provider.
Skill Requirement	Requires professional developers, testers, and system administrators.	Accessible to non-technical users, entrepreneurs, or small teams.

7. RESEARCH METHODOLOGY

My study follows the following Research Onion model by Saunders

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.
Data Collection	Structured observation and prototype performance logs	Ethically compliant and objective data sources.

To ensure methodological coherence, my study follows the Research Onion model by Saunders, which provides a structured, layered framework for research design. This structured approach ensures logical progression from philosophical foundations to practical execution.

8. PROJECT DESIGN OVERVIEW

Functional Requirements:

- ✓ **The system shall enable guards to complete and submit daily checklists digitally.**
- ✓ **The system shall allow guards to upload photographic evidence for verification.**
- ✓ **The system shall timestamp each task automatically upon completion.**
- ✓ **Supervisors shall be able to review, comment, and approve submitted reports.**
- ✓ **Data shall be securely stored in a centralized cloud database.**

The foundation of the design phase was built on operational observations of SME retail security environments. The requirements were derived from the need to improve daily reporting efficiency, ensure traceability of security tasks, and facilitate better communication between guards and supervisors

8.1 SYSTEM ARCHITECTURE

Modular and layered architecture comprised of three key modules:

✓ **GUARD MODULE (MOBILE APPLICATION)**

– Allows guards to perform digital checklists, upload photos, and add task notes.

✓ **SUPERVISOR MODULE (WEB DASHBOARD)**

– Provides managers with oversight of guard submissions and performance analytics.

✓ **DATABASE MODULE (CLOUD INTEGRATION)**

– stores all entries securely in DB collections, ensuring real-time synchronization.

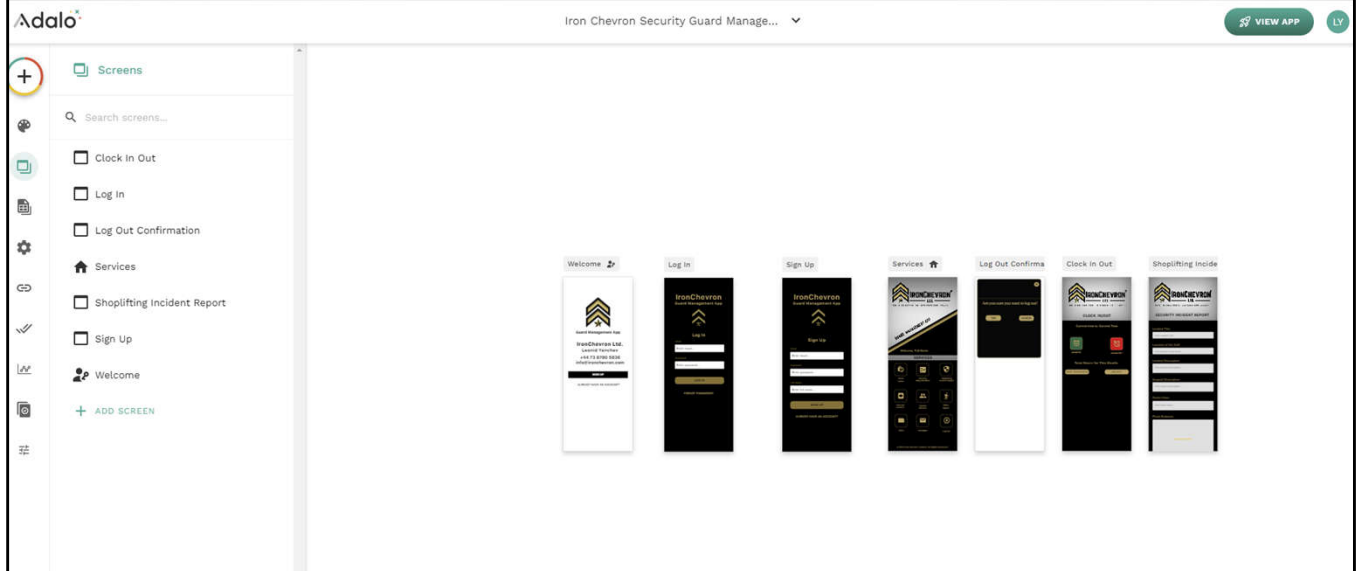
The project employed a modular and layered architecture to ensure maintainability and scalability. The system comprises three key modules:

- 1. Guard Module (Mobile Application)** – allows guards to perform digital checklists, upload photos, and add task notes.
- 2. Supervisor Module (Web Dashboard)** – provides managers with oversight of guard submissions, review options, and performance analytics.
- 3. Database Module (Cloud Integration)** – stores all entries securely in DB collections, ensuring real-time synchronization.

This modular approach supports separation of concerns, enabling each component to function independently while maintaining full system integrity

8.1 SYSTEM ARCHITECTURE-cont.

✓ GUARD MODULE (MOBILE APPLICATION)



Guard Module (Mobile Application) – allows guards to perform digital checklists, upload photos, and add task notes.

8.2 SYSTEM ARCHITECTURE-cont.

✓ DATABASE MODULE (CLOUD INTEGRATION)

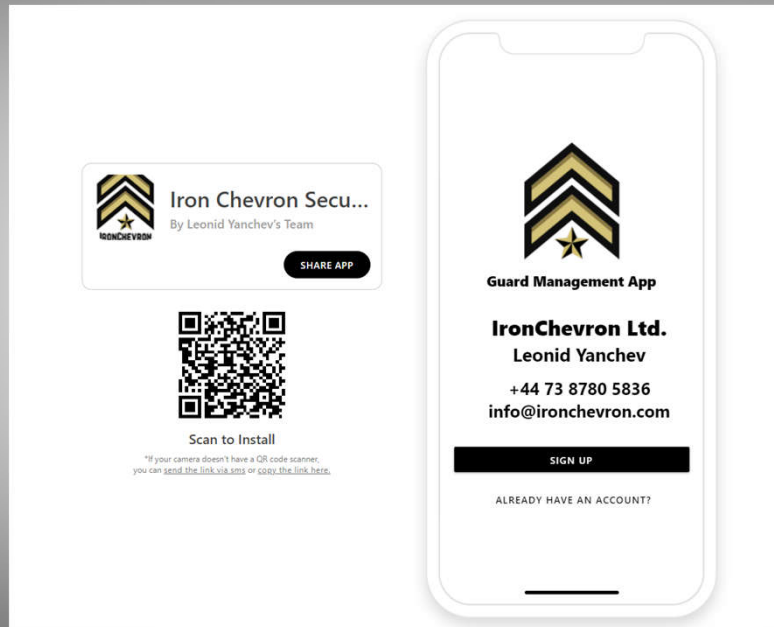
The screenshot displays the Adalo application interface for 'Iron Chevron Security Guard Manage...'. The left sidebar shows a list of database collections: 'Users' (5 Records), 'ClockRecords' (0 Records), 'DailyChecklists' (0 Records), and 'Shoplifting Inci...' (3 Records). The 'Shoplifting Inci...' collection is selected, and its fields are listed: Incident Title, Date & Time, Location of the theft, Incident Description, Suspect Description, Stolen Items, and Photo Evidence. The main area shows a table titled 'Shoplifting Incident Reports' with a '+ ADD SHOPLIFTING INCIDENT REPORT' button. The table contains three records:

Incident Title	Date & Time	Location of th...	Incident Descr...	Suspect Descr...	Stolen Items	Photo Evid...
Womenswear clothing ...	October 28, 2025 1:01 ...	Lower Ground Floor	IC-2 Female in her 20's...	IC-2 Female in her 20's...	1 x Under Armour T-shirt	
Jewellery theft	October 28, 2025 12:47...	Upper Ground Floor	IC-5 Female was brow...	IC-5 Female in her mid...	Necklace "Michael Kors"	
Handbag theft	October 28, 2025 12:37...	Upper Ground Floor	IC-1 male in his 40's in...	IC-1 male in his 40's	Black "Norh Pole" ruck...	

A 'DONE' button is visible at the bottom right of the table.

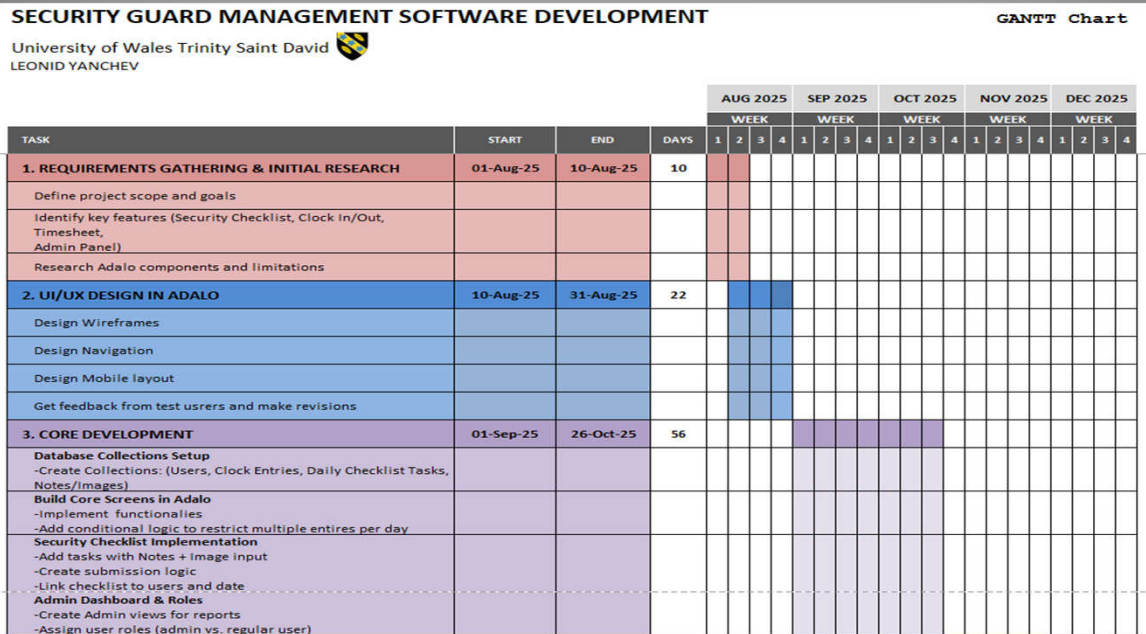
Database Module (Cloud Integration) – stores all entries securely in DB collections, ensuring real-time synchronization.

8.3 DEMONSTRATION



Instead of present you pictures of the prototype of the app I will demonstrate how it works. You can actually scan the QR code, and it will take you to this screen and you have to SIGN-UP with fictitious e-mail and password (it can be 123 or else) and a full name (also fictitious). Once you sign-up it will take you to the main screen SERVICES and from there just CLOCK-IN and then back to SERVICES and click on SHOPLIFTING INCIDENT REPORTS and once you fill in click CREATE SHOPLIFTING INCIDENT REPORT. Then I will log-on as Admin and I will present how 3 separate cloud DBs were filled in real-time with these information.

9. PROJECT TIMELINE (GANTT CHART)



“The project began on 1st August 2025 with the research and literature review phase, which concluded by mid August. By mid-October, I had completed the design and requirements phase, and as of today, 27th October, I’m in the implementation stage, developing the mobile and web components. The next phase, testing and evaluation, is planned to start on 23rd November, followed by report writing and final viva preparation before the final submission and presentation on 19th December.

9. PROJECT TIMELINE (GANTT CHART-cont.)

SECURITY GUARD MANAGEMENT SOFTWARE DEVELOPMENT

GANTT Chart

University of Wales Trinity Saint David
LEONID YANCHEV

[illegible]

10. CONCLUSION

In summary:

- ✓ The project is on track to deliver a functional, no-code SGM software.
- ✓ The research and design phases are complete. Implementation phase is in progress.

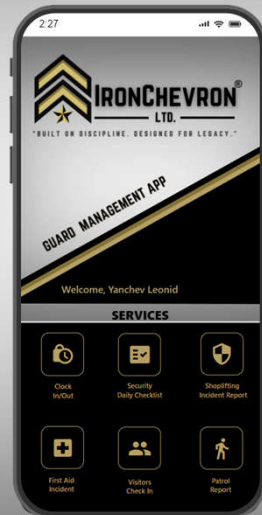
Key outcomes so far:

- ✓ A validated set of functional and non-functional requirements.
- ✓ A working prototype that demonstrates digital checklist completion in real-time.
- ✓ A clear roadmap for testing and evaluation.

Next Steps:

- ✓ Complete system testing (usability, reliability, and performance).
- ✓ Conduct performance evaluation using test metrics.
- ✓ Prepare the final dissertation and demonstration package.

THANK YOU FOR YOUR ATTENTION !



INTERIM VIVA PRESENTATION

LEONID YANCHEV

29-OCT-2025

