SECURITY GUARD MONITORING SYSTEM

By

Lee Jie Lun

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: SECURITY GUARD MONITORING SYSTEM

Academic Session: JAN 2024

I LEE JIE LUN

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:
21, Jalan Impian Utama,
Taman Impian, 81500,
Pekan Nanas, Johor.

TAN TEIK BOON
Supervisor's name

Date: 22/4/2024

Date: 25/4/2024

FACULTY/INSTITUTE* OF ____INFORMATION AND COMMUNICATION TECHNOLOGY_____

UNIVERSITI TUNKU ABDUL RAHMAN

Date: _22/4/2024___

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that _____**LEE JIE LUN**_____ (ID No: __**21ACB01925**__) has completed this final year project/ dissertation/ thesis* entitled "_____**Security Guard Monitoring System**_____" under the supervision of _____**Ts Tan Teik Boon**_____ (Supervisor) from the Department of ___**Computer Science**_____, Faculty/Institute* of ___**INFORMATION AND COMMUNICATION TECHNOLOGY** _____

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

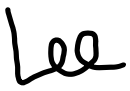
Yours truly,

_____

(*LEE JIE LUN*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**SECURITY GUARD MONITORING SYSTEM**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : LEE JIE LUN

Date : 22/4/2024

ACKNOWLEDGEMENTS

I would like to express thanks and appreciation to my supervisor, Ts Tan Teik Boon and my moderator, Dr Siti Nurlaili Binti Karim who have given me a golden opportunity to involve in an Security Guard Monitoring System. It is my first step to establish a career in this system. A million thanks to you.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

This report describes the creation of a Security Guard Monitoring System with the goal of increasing the efficiency and dependability of security operations. In a world where security is vital, this system employs technology to assure the proper monitoring and coordination of security personnel during patrols. The traditional method of manual recording, which used notebooks at checkpoints, has been shown to be vulnerable to tampering and to lack real-time reporting capabilities. This solution tackles significant concerns that conventional and existing security guard monitoring systems encounter, such as tampering, missing patrols, and a lack of centralized control. To minimize tampering, the system will use Near Field Communication (NFC) technology at checkpoints, replacing traditional notebook with NFC tags. NFC technology can capture data in real time, decrease the risk of tampering. Additionally, log movement will be used to reduce the number of missing patrols by immediately alerting the system manager in the event of a deviation. Furthermore, this system integrates current existing infrastructure to provide a centralized control system. This integration will improve security operations by requiring security staff to follow predetermined routes and regulations. This system will be built with React Native framework to provide cross-platform capabilities.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

NFC	Near Field Communication
FCM	Firebase Cloud Messaging

CHAPTER 1 INTRODUCTION

In an era where security concerns take precedence, the effective supervision and coordination of security personnel become indispensable in safeguarding individuals and assets. The Security Patrol Monitoring System stands as a significant technological advancement in the region of security management This comprehensive solution, encompassing both software and hardware components, is crafted with the primary objective of seamlessly tracking and overseeing the movements of security guards and other security personnel during their patrols in certain areas. This system is positioned to ensure security personnel adherence to prescribed patrol routes, checkpoint protocols, and facilitating rapid responses to emergency scenarios.

There is a traditional method for documenting the activities of a security guard, involving the use of a notebook placed at each checkpoint within the patrol area. In this method, security guards are required to manually record their information such as date, time, name, and status of the area each time they pass by the checkpoints. It is undeniably straightforward and budget-friendly for monitoring the security guard's movements. A significant drawback of this notebook log is the time-consuming nature of the verification process for supervising personnel. Furthermore, it lacks the capability to offer real-time reporting, which can be particularly problematic in situations where a security incident occurs at a nearby checkpoint. In such cases, there can be delays in notifying others for assistance. In addition, it was easy to be tampered as well. Security personnel may write multiple record with any times on the notebook. To address these challenges, there is a critical need for a real-time system that can issue immediate warnings and promptly request assistance when required.

1.1 Problem Statement and Motivation

Security guard has performed a role of protecting our lives and asset. Such that, an efficient security guard monitoring system is important. There are still facing challenges in this system even it has already developed numerous modern monitoring systems in this region. Below will present three critical issues faced by both traditional and current security guard monitoring systems: tampering, missing patrols, and the absence of centralized control.

One of the primary challenges in traditional manual patrol monitoring methods and current security guard patrol monitoring system is tampering. Security guards can employ various techniques to manipulate paper-based logbooks or QR code in current system, allowing them to provide false records of their movements. This issue not only compromises the accuracy of patrol data but also undermines the integrity of the entire security operation. False reports can conceal potential security breaches, making it essential to address this problem.

An additional challenge that can arise within both traditional and current systems is the missing patrols. Whether due to negligence or unforeseen events, missing of patrol coverage can occur, resulting in critical areas being left unattended. Danger can arise at any time and any place. Therefore, it is crucial to have a real-time system that can identify when there are no security personnel present in a designated area once the allotted time has expired. These lapses have the potential to compromise overall security and require proactive measures to be taken for resolution.

The absence of centralized control and coordination remains a challenge in many current security guard patrol monitoring systems. Most of the current system is difficult to integrate the system with their existing infrastructure. For example, supervising personnel cannot access control to the surveillance camera which implemented at the checkpoint via the system. These systems often suffer from fragmented and dispersed information spreading across multiple platforms or databases, which makes it difficult to access and coordinate effectively when it is most crucial.

The challenges of tampering, missing patrols, and the absence of centralized control happened in both traditional and current security guard patrol monitoring systems. Current system has implemented various advanced technology like biometric authentication, NFC, or real-time database system to solve the problem. These

advancements mark a significant progression in security operations, guaranteeing the well-being and safeguarding of individuals and assets within a constantly changing security environment.

For the motivation, in the region of security guard patrolling, effective information management is crucial for gathering valuable data. The integrity of this data plays a vital role in ensuring its usefulness. Besides, we have to ensure the security personnel is patrolling properly to ensure the safety of people and asset. Without proper management and safeguarding, collected information may become useless. Therefore, the objective is to develop a real-time security guard monitoring system that minimizes tampering issues and provides a range of tools to facilitate an efficient and user-friendly management process. This feature also serves to enhance the safety of security personnel as the system operates in real-time. Consequently, they can promptly request assistance or notify individuals to steer clear of specific areas.

1.2 Objectives

Objective 1: Implement NFC technology to reduce tampering

To deal with the tampering issue in manual recording, the use of NFC technology can obviously reduce the risk of the system being tampered. The system implements the NFC tags at every checkpoint to retrieve essential information like date, time, location, name of security personnel and current situation at the location. By implementing NFC technology, the system aims to restore trust in patrol records, enhancing the overall reliability of security operations.

Objective 2: Implement log movement to minimize missing patrol

To minimize missing patrol, the system implement log movement to get the information in real time. The system get the information by NFC technology. The system will instantly alert the system manager if there is any deviation or missed checkpoints. By minimizing instances of missing patrols, the system enhances security coverage, providing peace of mind to organizations and individuals alike.

Objective 3: Integrate with existing infrastructure

There is lack of centralized control of existing infrastructure in most of the current system resulting in fragmented information and inefficient management. The third objective aims to develop a system that integrates several features into the existing infrastructure. By centralizing the control into a system, it provides ease to system manager by accessing security alert, generating reports in real time.

1.3 Project Scope and Direction

The project scope is to create a digital security guard patrol monitoring system by implementing NFC technology to reduce the risk of tampering. Log movement will also be implemented to minimize the instances of missing patrol to track the movement of security personnel. The system will also integrate with existing infrastructure into one centralized system to ease the work of system manager. The system is using React Native to develop due to its cross-platform functionality.

1.4 Contributions

The Security Guard Monitoring System enhances security by transmitting patrol data through the internet, enabling real-time reporting and alarm monitoring. This innovative system employs a pair of NFC reader and writer (normally a phone integrates with NFC and a NFC tag), which NFC tags strategically placed at key locations within the client's facility or premises. As security guards conduct their patrols during their assigned hours, using their phone to interact with the tag to captures relevant data, including timestamps and observations. This can reduce the risk of tampering.

Security guard monitoring system has the ability to relieve security managers from the arduous task of assigning responsibilities and simultaneously overseeing multiple areas. The security manager in the office takes charge of determining which guards will participate in a guard tour, scheduling their patrols, defining patrol locations, establishing checkpoint locations, and ensuring that they are appropriately scanned or identifying any potential missed checkpoints. This streamlined approach allows for well-organized guard visits, ensuring that security and safety standards are consistently upheld and that no assets are put at risk.

This data is accessible via any computer with an internet connection, allowing for comprehensive monitoring of all security guard activities and patrol checkpoints. To further enhance the efficiency of the Security Guard Monitoring System, security companies have the option to integrate it with surveillance cameras that implement at every checkpoints. This integration adds an additional layer of security and ensuring a more robust security infrastructure. [5]

1.5 Report Organization

In this report, there are total of seven chapters which includes introduction of the project, literature review, system methodology/approach, system design, system implementation, system evaluation and discussion, and conclusion and recommendation.

In the chapter 1 introduction, it explains about the importance of Security Guard Monitoring System and state out clearly the challenges in current existing security guard monitoring system. Furthermore, this chapter talks about the objectives which list out the specific goals to be achieved. Project scope and contributions is discussed as well to clarify the improvement this system can bring out compared to current existing system.

In chapter 2, it first explores on the similar system to the proposed system and find out the possible advancement that can be learnt and the potential issue that the system could have. QR-Patrol, eSmartGuard and a wireless control system based on Bluetooth and IBeacon technology has been reviewed in this chapter. Then, a critical remark has been done to compare each system. At last of this chapter will be the research on technologies which includes NFC and React Native framework.

Chapter 3 present the proposed methodology, outlining the approach that support the security guard monitoring system. System design diagram and system architecture diagram like ERD and use case diagram will be on this chapter as well. Then, it will finally outline the project timeline.

Chapter 4 of this report dive into the system design of this project. It starts with the illustration of system block diagram followed by the specification of the each components and modules shown in the diagram. Then, it followed by a more detailed section that outlines the flows of operation in each component and its explanation in system component interaction operations.

Chapter 5 starts with the specification of hardware and software setup that required for developing and testing this system. Then, it followed by the configuration to setup the coding environment. Next, it shows the overall system operation with screenshot provided. Each screenshot indicating each important module. The

challenges and issue met is outlined in this chapter as well to show the difficulties when developing this system and what is the feature failed to implement.

In Chapter 6 of this report, testing and evaluation of the developed system are conducted. Each test case is detailed with its test case name, precautions, step-by-step execution process, and expectation, with accompanying screenshots demonstrating the system's performance. The outcomes of these tests, whether they passed or failed, are recorded along with observations drawn from the results. Additionally, project challenges encountered during entire project development are highlighted. Furthermore, the chapter evaluates the system in achieving its objectives by comparing the defined project objectives with the implemented functionalities of the system, assessing how well it addresses the identified problem and fulfils the project's goals.

Last but not least, the chapter 7, conclusion is just a briefly explain like a summary of this report on what has been done. Then it followed by the recommendation which will highlight the future work and improvement can be done for this system. And after that are the references weekly report and plagiarism check result to serve as appendix.

CHAPTER 2 LITERATURE REVIEW

2.1 Previous Works on Security Guard Monitoring System

2.1.1 QR-Patrol [1]

QR-Patrol [1] is first launched in year 2013 and it is a security guard patrol monitoring system that revolutionizes how security professionals manage and monitor their patrol operations. Its aim is to developed a real-time security guard patrol monitoring and managing system. Designed to enhance security, accountability, and efficiency, QR-Patrol utilizes cutting-edge technology to streamline patrol activities and ensure real-time tracking and reporting.

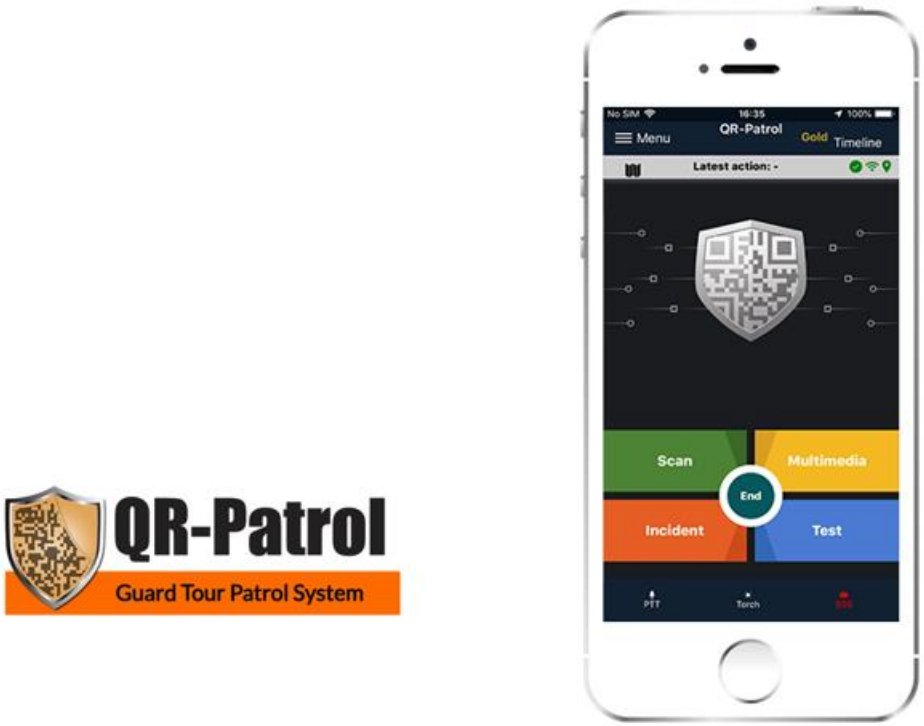


Figure 1 QR-Patrol Application

Each security guard on a patrol now utilizes cutting-edge technology, enabling them to efficiently manage their tasks using a smartphone while maintaining complete control over incident responses. The process of inspections and guard tours is significantly improved, allowing guards to attach images, voice messages, notes, and signatures to scanned items. Furthermore, security personnel, including guards and other workers, have the capability to instantly notify facility managers responsible for monitoring assets and locations.

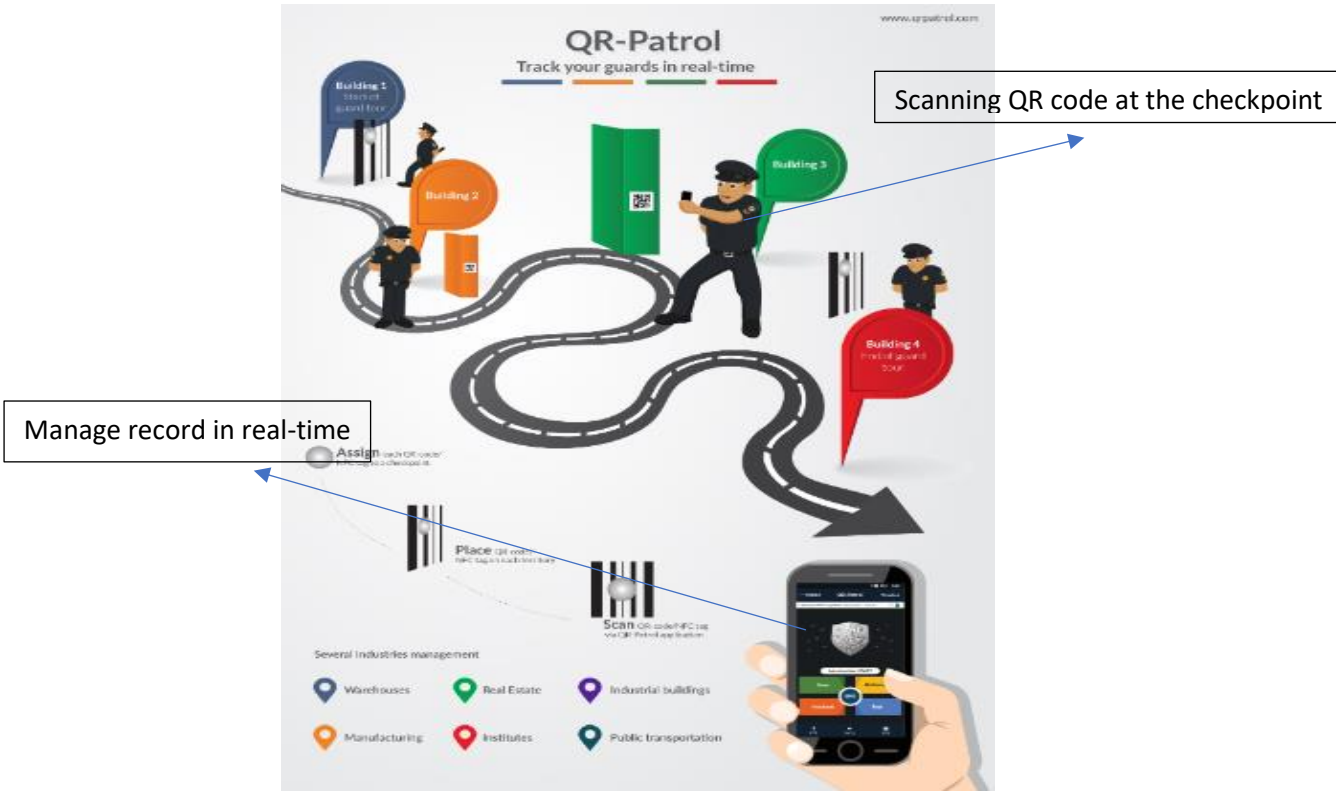


Figure 2 QR-Patrol Overview

QR-Patrol has introduced an innovative system where QR codes are positioned at each checkpoint for security personnel to scan. After scanning, it instantly transmits useful information to the QR-Patrol Web Application via a cloud server. This significantly reduces the workload required for verifying the security guard's performance. However, this approach does not entirely eliminate the possibility of deception, as security personnel can capture a photograph of the QR code and scan it whenever they want to avoid actual patrolling.

Features:

- QR code scanning at checkpoints
- Various media options available for uploading to ensure accurate information within the management system.
- Ability to instantly inform facility manager about the current situation

Cons:

- QR code scanning technique is easy to be tampered

- Cannot operate without internet connection

Suggested way to improve:

- Replace QR code with NFC tags
- Use SMS to record when there is no internet connection

2.1.2 eSmartGuard [2]

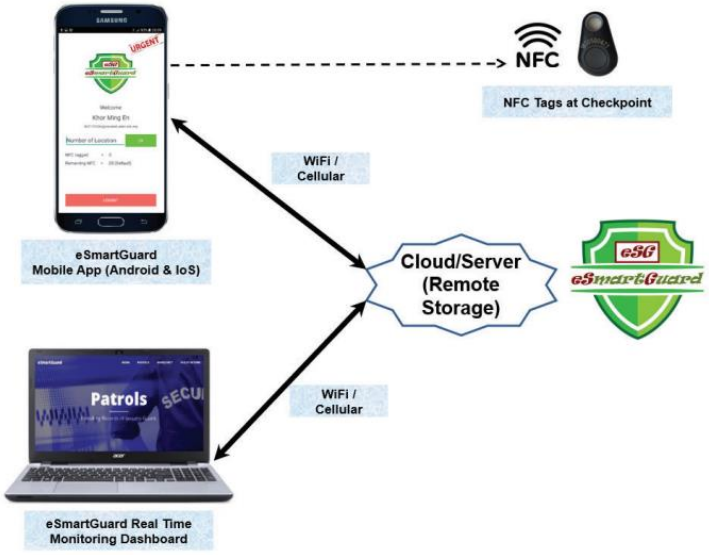


Figure 3 eSmartGuard Architecture

The eSmartGuard [2] system comprises three core components: the Guard Mobile App, the Superior Mobile App, and the Administrative Hub known as the eSmartGuard Web Dashboard App. Both the mobile apps and the web application are seamlessly connected to a cloud server using either WiFi or cellular networks. In addition, NFC tags are strategically placed at various checkpoints to facilitate the patrol procedures.

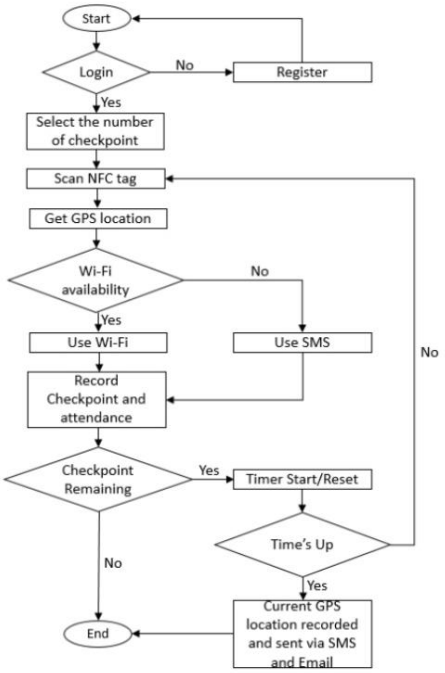


Figure 4 eSmartGuard System Flowchart

To deal with QR code tampering problem, NFC has been used widely used in security guard patrolling system. eSmartGuard [2] has introduced an alternative approach, replacing the QR codes typically placed at checkpoints with NFC tags. In this system, security guards carry an NFC reader, often a smartphone with NFC capability, which they use to scan the tags at each checkpoint. Additionally, eSmartGuard includes a timer feature that triggers a warning when a predefined time limit expires, ensuring the safety of security personnel. However, it's important to note that eSmartGuard lacks integration with existing infrastructure, such as surveillance cameras, which could offer additional support and insights for patrolling activities.

Pros:

- Timer features to ensure the safety of security personnel
- NFC is harder to be tampered
- Can use SMS to record when there is no Wi-Fi availability

Cons:

- Did not integrate with existing infrastructure
- Without ability to post different type of medias to the system

Suggested way to improve:

- Integrate with surveillance cameras at checkpoints to provide better view on the location when emergency occurred
- Make the system accept different types of medias like voice, images.

2.1.3 A WIRELESS CONTROL SYSTEM BASED ON SMART BLUETOOTH AND IBEACON TECHNOLOGY FOR AUDITING THE PATROLS [3]

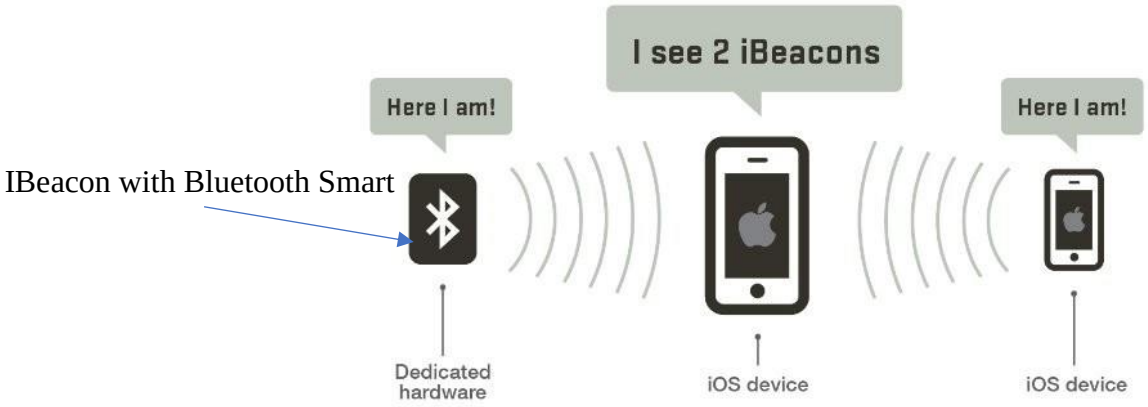


Figure 5 Bluetooth and IBeacon

In this type of security guard patrol monitoring system, there are 2 main components which are Smart Bluetooth and IBeacon. Bluetooth 4.0 introduces two device categories which are Bluetooth Smart and Bluetooth Smart Ready. iBeacon is a wireless communication protocol that Apple introduced to complement Bluetooth Smart technology. These beacon devices emit signals at regular intervals, allowing mobile applications to ascertain their location based on the information transmitted by these beacons. In this system, it implements Bluetooth Smart Ready devices (normally tablet or smartphone) with Bluetooth Smart device which is IBeacon.

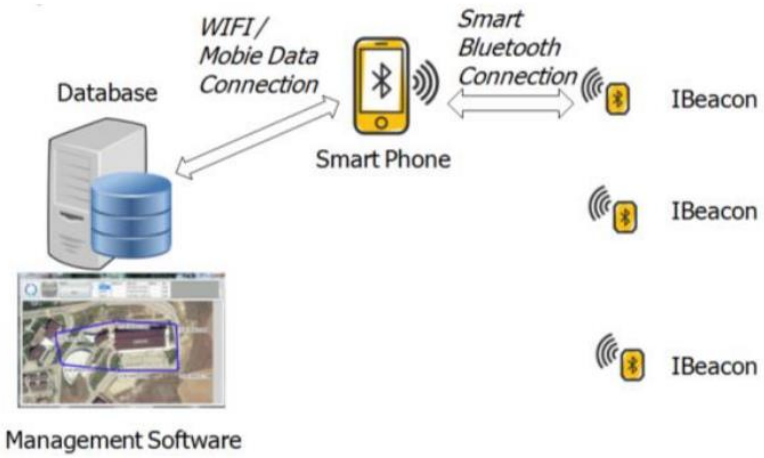


Figure 6 Architecture of the system

Every security guard is equipped with a smartphone containing a dedicated mobile application. This app employs Smart Bluetooth Ready technology to scan its surroundings for signals emitted by iBeacon devices. Upon detecting a beacon signal, the mobile application deciphers it to identify the iBeacon's unique ID and estimate its distance from the smartphone. Subsequently, the app compiles a log entry, encompassing the identification numbers of both the cell phone and iBeacon, the timestamp, and the iBeacon's proximity. This log entry is then transmitted and stored in a centralized database. The management software is responsible for generating the requisite reports based on this collected data.

Pros:

- Cheap and more energy-efficient

Cons:

- Do not ensure the safety of security guards

Suggested way to improve:

- Add warning functions to inform security personnel nearby for help when emergency occurred

2.2 Critical Remarks

System / Features	QR-Patrol	eSmartGuard	[3]
QR code	✓		
NFC		✓	
Bluetooth 4.0			✓
Cheap and energy-efficient			✓
Timer feature to ensure security guard safety		✓	
Integrate with existing infrastructure	✓		
Ability to accept different types of medias	✓		
Notify security personnel instantly when emergency occur	✓	✓	
Can operate without internet connection		✓	
Avoid tampering		✓	✓

Table 2.1 Critical Remarks

2.3 Reviews of Technologies

2.3.1 NFC

Near Field Communication (NFC) is a wireless communication technology designed for short-range interactions. What makes NFC powerful is its simplicity—communication is initiated by a simple touch between devices within a short distance, and it ends instantly when the devices are separated. A fundamental advantage of NFC lies in its security, attributed to its limited communication range. In NFC communication, bringing two devices into very close proximity initiates communication, while moving the devices apart beyond a specific range instantly terminates the communication [4].

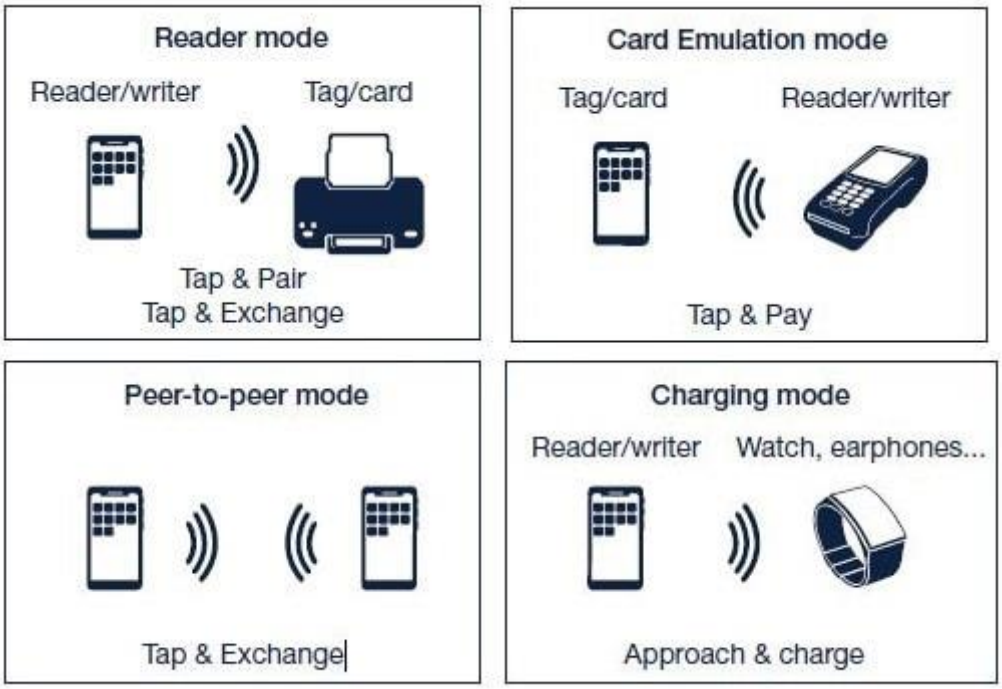


Figure 7 Categories of NFC

NFC communication takes place when two NFC-compatible devices are positioned within close range, typically a few centimeters apart, and operates at a frequency of 13.56 MHz (as shown in Figure 2.) [4]. NFC facilitates seamless communication among a variety of NFC devices, adhering to ISO/IEC 18000-3 air interface standards, and offers transfer rates of 106, 212, and 424 Kbits per second.

2.3.1 React Native

React Native is a framework developed by Facebook in 2015. Normally developers need to develop at least 2 program to adapt their program in both Android and iOS platforms. However, since the basic code remain consistent, development process can use a single programming language for adjusting the visual presentation to the unique platform by using proper elements. Facebook once said that “learn once, write anywhere” which is mainly talking about React Native.

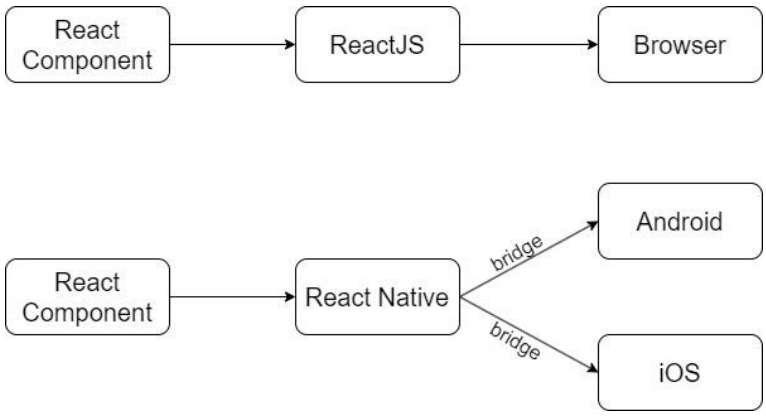


Figure 8 Rendering in React and React Native

Instead of executing React within a web browser and rendering elements like divs, text, React Native operate within a embedded instance of JavaScriptCore on iOS or V8 on Android, inside the application itself. It then renders content by using platform-specific, higher-level components. The rendering overview is shown in the Figure above. As previously stated, React Native can translate React Native components into native Views on Android or UI Views on iOS. This feature is enabled by an abstraction layer known as the "bridge." This bridge allows React Native to use the rendering APIs specific to Java for Android or Objective-C for iOS. [9]

CHAPTER 3 SYSTEM METHODOLOGY/APPROACH

3.1 Methodology

Prototyping is used for the development of this system. It came as a process in response to the need for more exact specification's definition. It includes developing a demonstration version of the software product that has key functionalities. A first version of prototype will be developed based on the initial specifications given. This first version is for further refinement since it serves as a foundation of how the system look like. The prototype keep updating and getting feedback until the prototype is satisfying, then it can proceed to further development stages. [8]

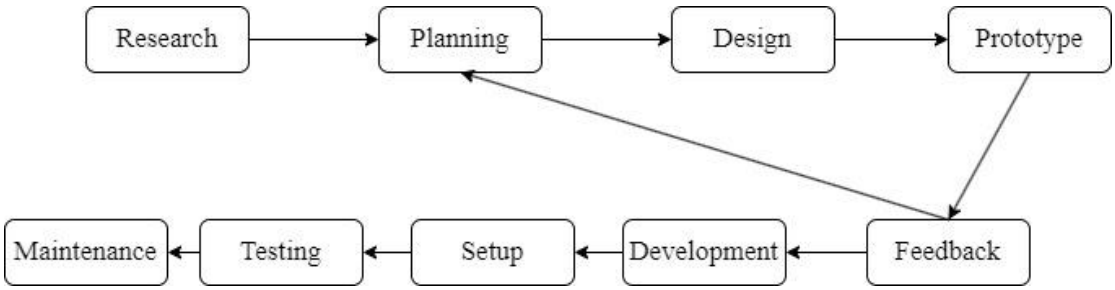


Figure 9 Prototyping process

The initial phase is the research phase. In this phase, I conducted research on three similar systems: QR-Patrol, eSmartGuard, and a wireless control system utilizing smart Bluetooth and iBeacon technology. I have outlined the advantages and disadvantages of each of these systems. My intention is to draw insights from the strengths of these systems and address the weaknesses to create an improved system.

After the research phase comes the planning phase. During this step, I will compile a list of system requirements based on the problems found in the current security guard monitoring system. The requirements has been prioritized based on their importance and feasibility, it is to ensure that the most critical issue has been solved. Then, I will create a timeline to record down the milestones of this system. In the design phase, these proposed requirements will be utilized to develop a comprehensive system overview. This includes the design of elements like user interface and business logics.

In the prototype phase, I will use Figma to create a high-fidelity prototype. During the development and implementation phases, this advanced prototype will

provide a detailed preview of how the system performs and behaves. For example, the prototype does not only show the system navigation and event handling, it also simulate user interaction and work flow. Refinement of prototype will be iterated for several times to best solving current system issue and matching system requirement before going into development phase. The prototype can be a valuable tool for usability testing.

During the development phase, having an advanced prototype as a guide simplifies the process significantly. My approach will involve initiating the coding process and seamlessly integrating all the essential features, including NFC technology and alarm alerts, into the system. Subsequently, the system, along with its corresponding hardware and software components, will be deployed and subjected to rigorous testing to ensure its optimal performance. Ongoing maintenance and support will follow to uphold the system's functionality and reliability.

Working on the development phase, advanced prototype simplifies the difficulties on coding out the system. I will develop actual code and integrate all the important features like NFC technology and alarm alert into the system. Subsequently, the system, along with its associated hardware and software components, will be deployed and tested to ensure optimal performance. Ongoing maintenance will be provided to ensure the system's functionality and dependability.

3.2 System Design Diagram

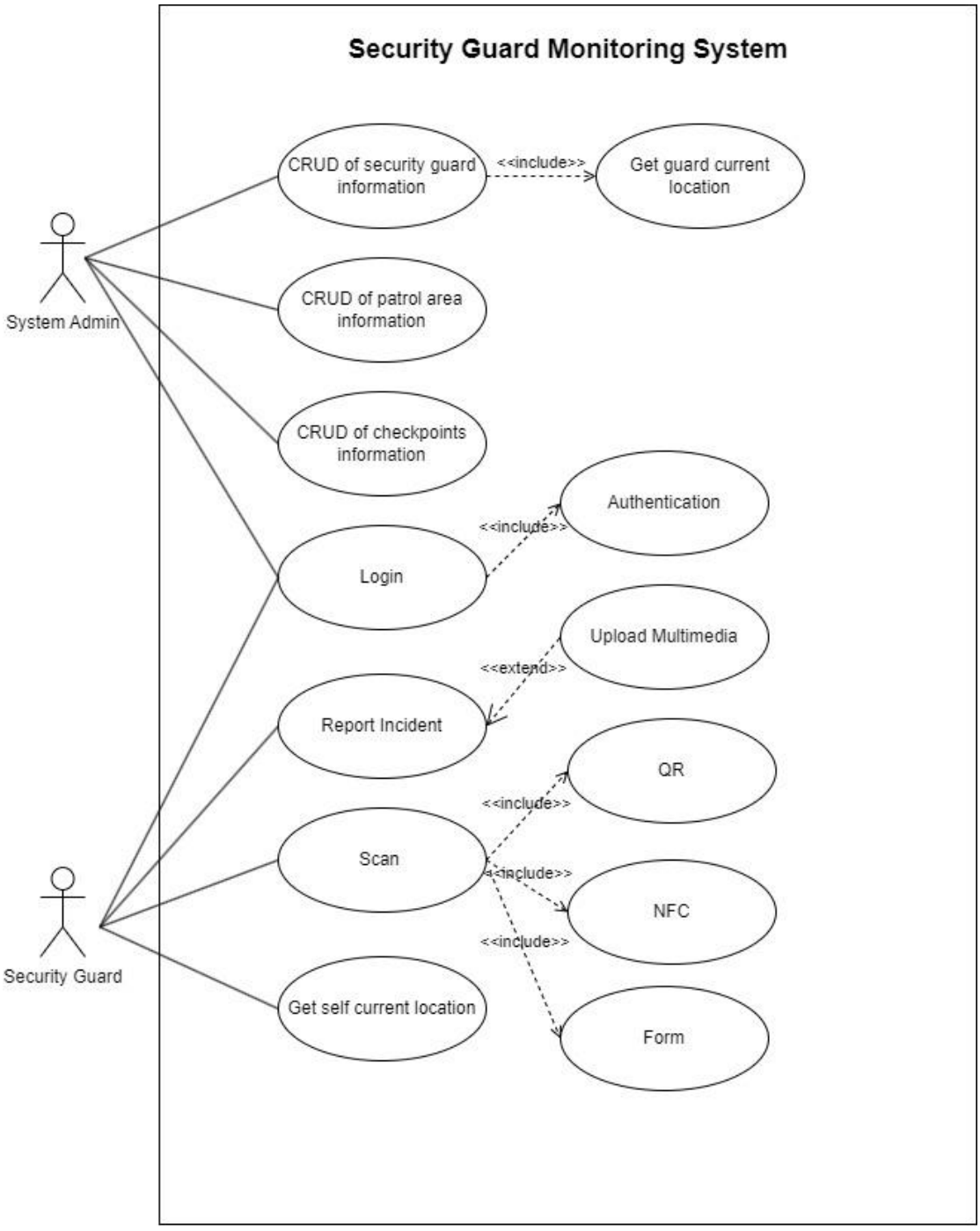


Figure 10 System Use Case Diagram

3.3 System Architecture Diagram

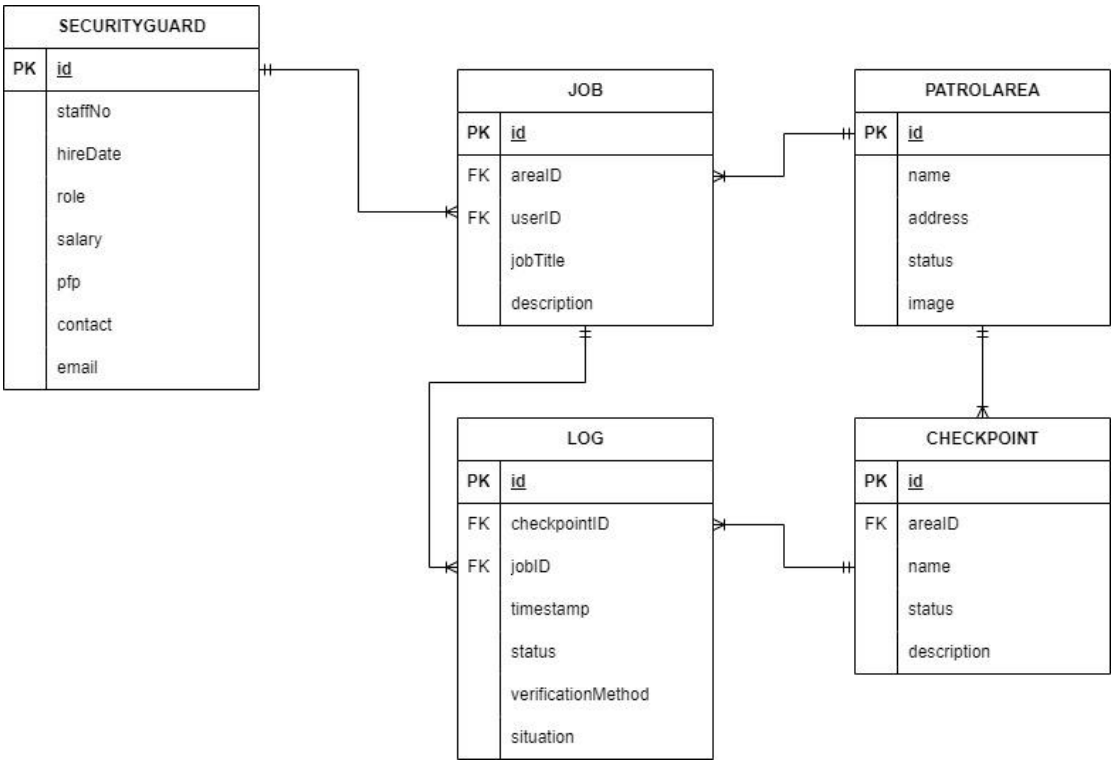


Figure 11 System Entity Relationship Diagram

The Entity Relationship Diagram (ERD) serves as a overall visual representation, clarify the included components within the system. The SECURITYGUARD entity represents the security guards patrolling specific regions under the eye of the system's administrator. JOB entity represents the jobs or assignments given to security guards to patrol specific areas. PATROLAREA represents the patrol areas that security guards are going to patrol. CHECKPOINT entity represents the checkpoints within each patrol areas where security guards must record their movements. LOG entity represents the record of security guard movements at specific checkpoints.

3.4 Timeline

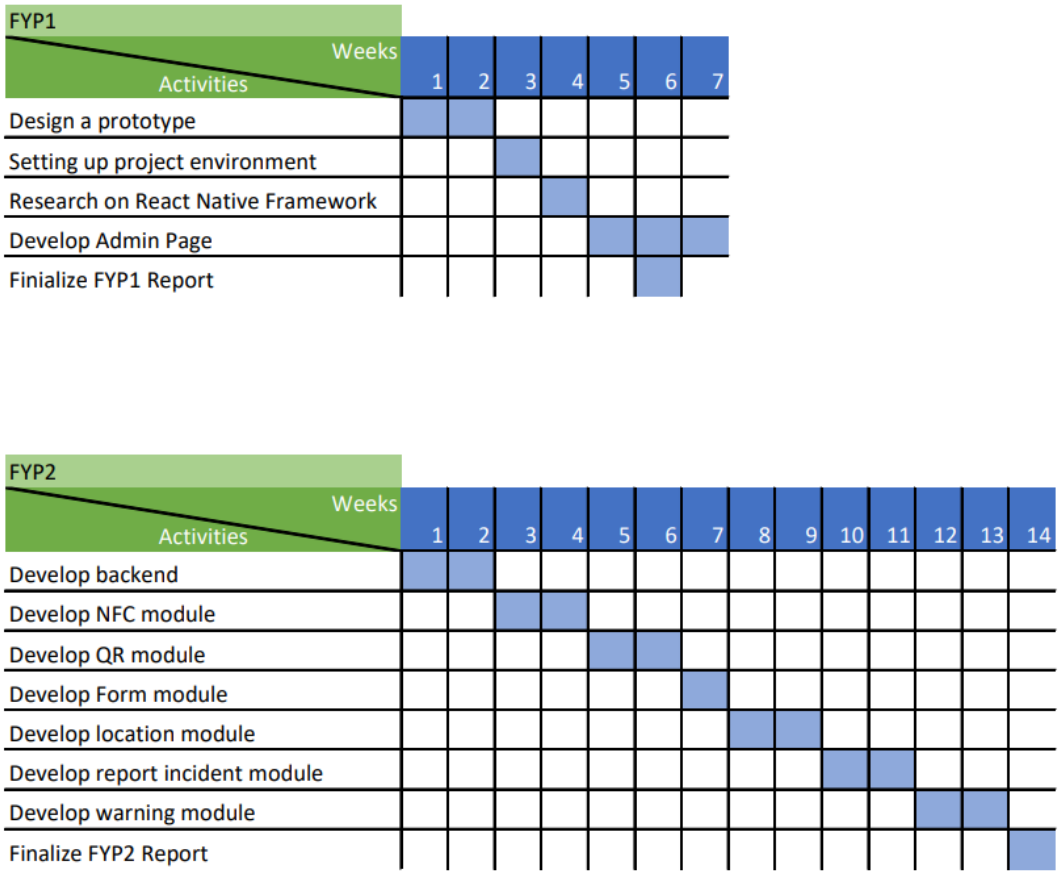


Figure 12 System Progress Timeline

CHAPTER 4 SYSTEM DESIGN

4.1 System Block Diagram

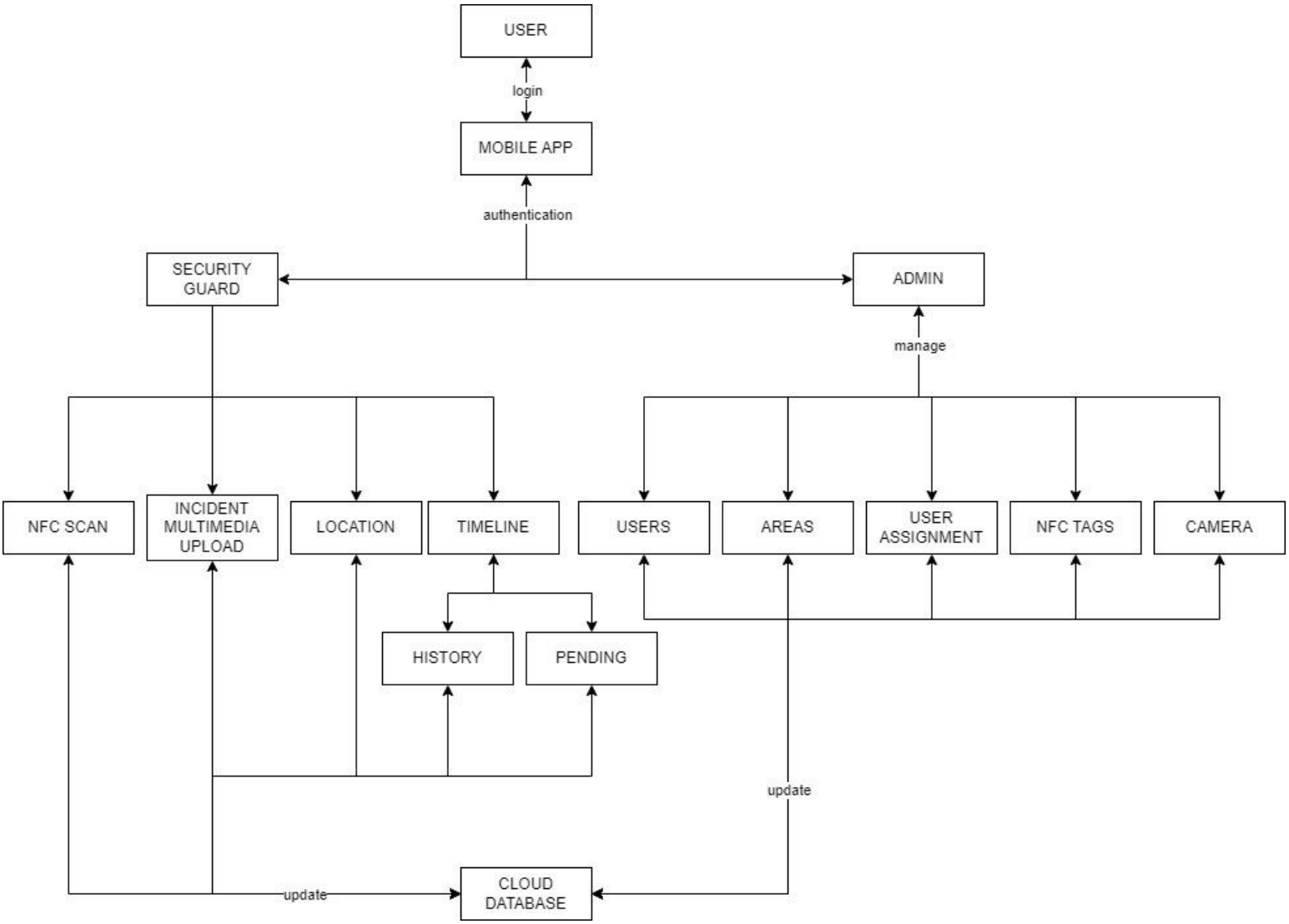


Figure 13 System Block Diagram

4.2 System Component Specifications

In this Security Guard Monitoring System includes two user roles: security guards and administrators. Security guards have access to specific modules important for their daily tasks. These include the NFC Scan module for tracking attendance through NFC tags, the Incident Multimedia Upload feature for recording unexpected incidents with multimedia evidence. Location module is to display real-time location information, and a Timeline module that shows both pending and completed assignments within the Pending and History section respectively.

On the other hand, administrators have wider range of management capabilities within the system. They can perform CRUD operations (Create, Read, Update, Delete) on key entities such as Users, Areas, Users' Assignments, and NFC Tags. The administrators can also watch the surveillance camera installed on each checkpoint in each area. This administrative functionality allows for comprehensive oversight and control over user accounts, geographical areas, task assignments, NFC tag management, and areas' situation.

All interactions and data updates performed within the system, whether by security guards or administrators, are seamlessly synchronized with a Cloud Database. This integration ensures centralized data storage, facilitating data consistency, accessibility, and scalability across the entire Security Guard Monitoring System.

The NFC Scan module implements an anti-tampering method to enhance data integrity by limiting scanning to NFC tags within a close range of 2 to 3cm from NFC-enabled mobile devices. This approach prevents security guards from using QR code scans to falsify attendance records. Each NFC tag installed at checkpoints contains encrypted checkpointId and areaId data using react-native-crypto-js, avoid to duplicated NFC tags with same information stored appear and ensuring system security. Additionally, the unique uid of each NFC tag is stored in the Cloud Database to ensure it is the real installed NFC tags at checkpoints. When a scan occurs, the attendance log is updated in the Cloud Database based on the checkpointId and areaId retrieved from the NFC tags. This integrated security mechanism guarantees accurate and secure attendance logging within the Security Guard Monitoring System.

The Incident Multimedia Upload module enables security guards to take or upload photos from their mobile devices directly to the Cloud Database. Each

uploaded image is labeled with predefined incidents such as fire alarm, break-in, water leak, or custom labels. Security guards are required to specify the area and checkpoint associated with the incident for accurate location tracking. This information enables administrators to ask for help for the security guard.

In the Location module, the system displays the security guard's current location on a map using the Google Maps API. The map points out the checkpoints assigned to the security guard, helping them navigate to each location efficiently. However, this feature is most effective for outdoor use as the Google Maps API may not accurately function indoors. In indoor settings where NFC scanning can be used, the Location module may not be applicable due to limitations in indoor mapping capabilities.

The Timeline module consists of two tabs: History and Pending modules. In the History tab, security guards can check logs of their checkpoint scans within assigned areas to ensure no checkpoints were missed. The Pending tab displays the security guard's current assignments, providing details about the next checkpoint they need to attend and the scheduled time for attendance. This feature helps security guards stay organized and informed about their upcoming tasks and responsibilities.

In the Users module, administrators can manage user roles by switching users between "user" and "admin" statuses. This determines which interface users access for specific tasks. Administrators also have the ability to delete users if they are found suspicious or are no longer working in the company.

In the Area module, administrators can create new geographical areas and following create checkpoints within those areas. Before deploying NFC tags at checkpoints, administrators can use this module to write encrypted checkpointId and areaId data onto the NFC tags. This simplifies the process and eliminates the need to manually input IDs into the NFC tags.

In the User Assignment module, administrators is allowed to create new assignments for users. This feature enables administrators to assign tasks to security guards without the need for face-to-face communication. Once an assignment is updated in this module, it will automatically appear in the security guard Timeline -> Pending tab. This streamlined process ensures that security guards are informed of

their assigned tasks and responsibilities without requiring direct communication from administrators to increase the efficiency.

The NFC Tags module is used to input and store the unique identifiers (UIDs) of NFC tags used at each checkpoint in the Cloud Database. Each UID stored in this module must be unique and authorized to update checkpoint logs, which helps prevent tampering. Even if the information of one NFC tag is copied onto another by a security guard, the system will verify the UID stored in the NFC tag UID table before updating the Cloud Database during NFC tag scans. This process ensures the integrity and security of checkpoint logging within the system. Additionally, the UIDs stored in the NFC tags are only readable, further enhancing the security of the system.

In the Camera module, administrators is allowed to connect to surveillance cameras that are installed at checkpoints within the system. This feature enables administrators to remotely access live video feeds and surveillance footage from cameras deployed at specific locations. By connecting to surveillance cameras through this module, administrators can acknowledge the current situation, view recorded footage, and ensure the security and safety of the areas. This functionality enhances overall surveillance capabilities and facilitates effective monitoring and management of security operations within the Security Guard Monitoring System.

4.3 Circuits and Components Design

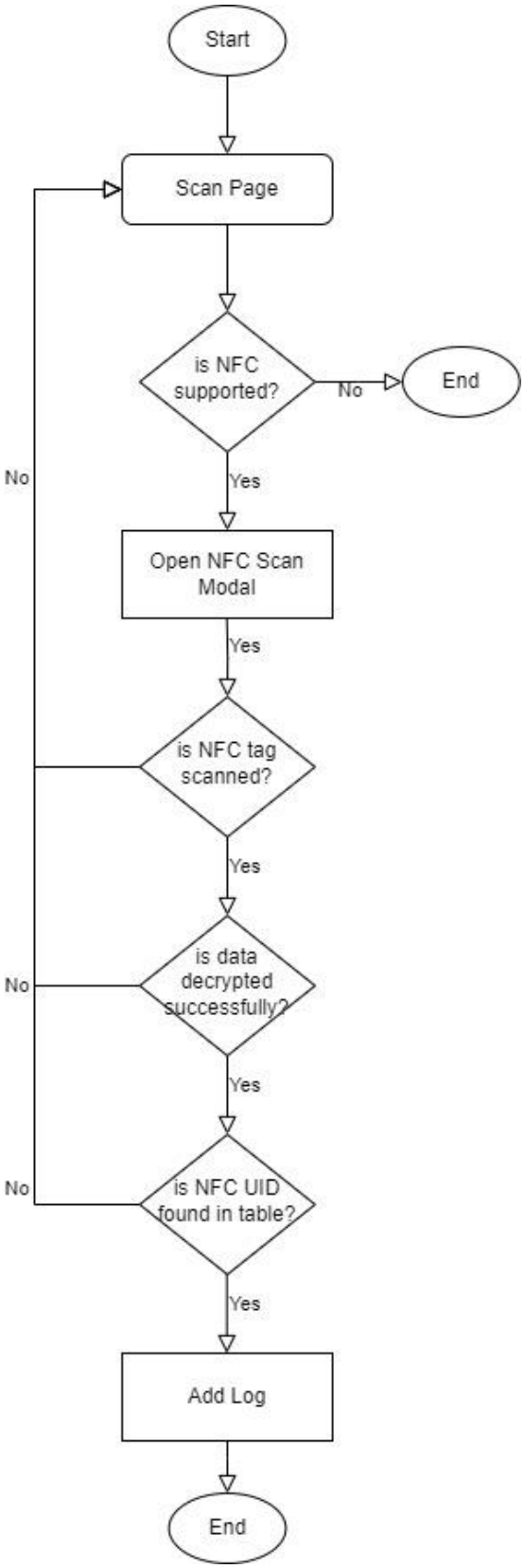


Figure 14 NFC Scan module flowchart

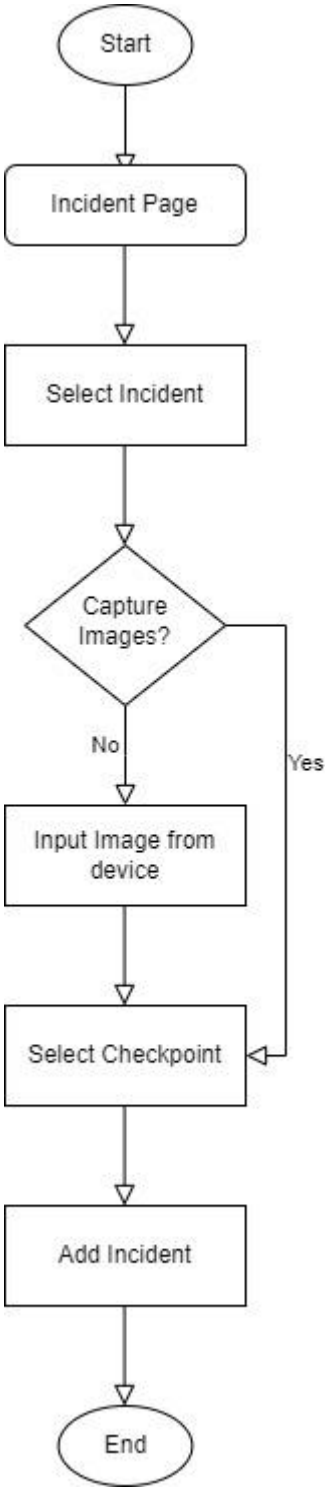


Figure 15 Incident module flowchart

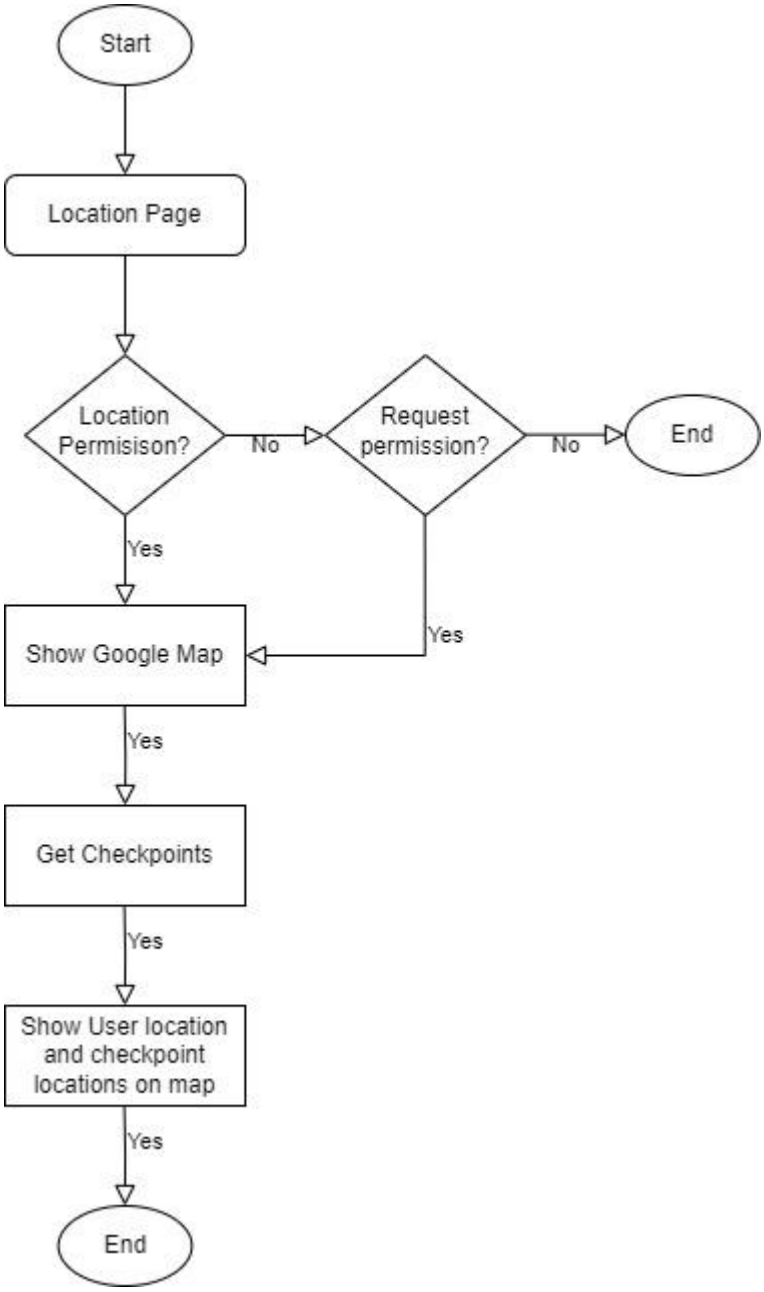


Figure 16 Location module flowchart

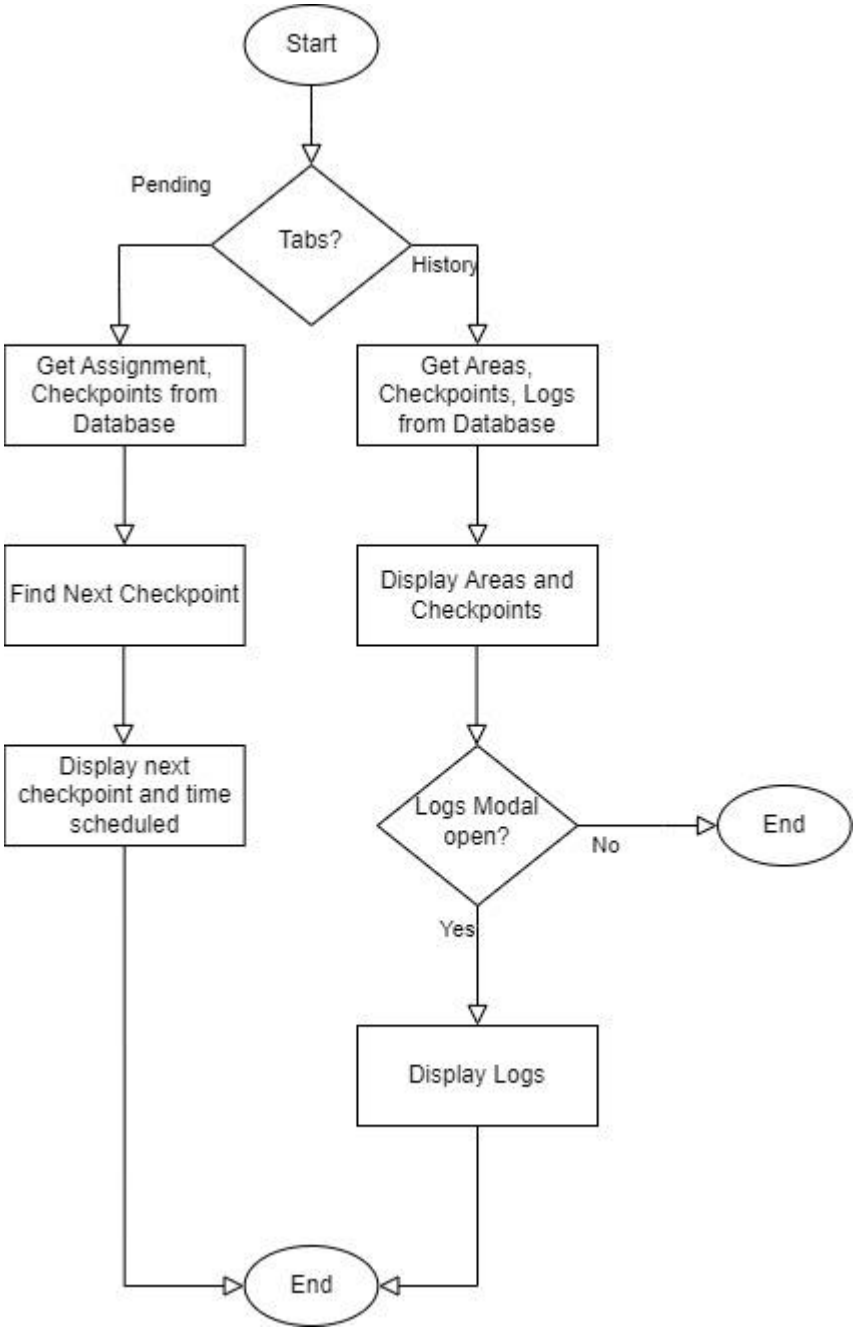


Figure 17 Timeline module flowchart

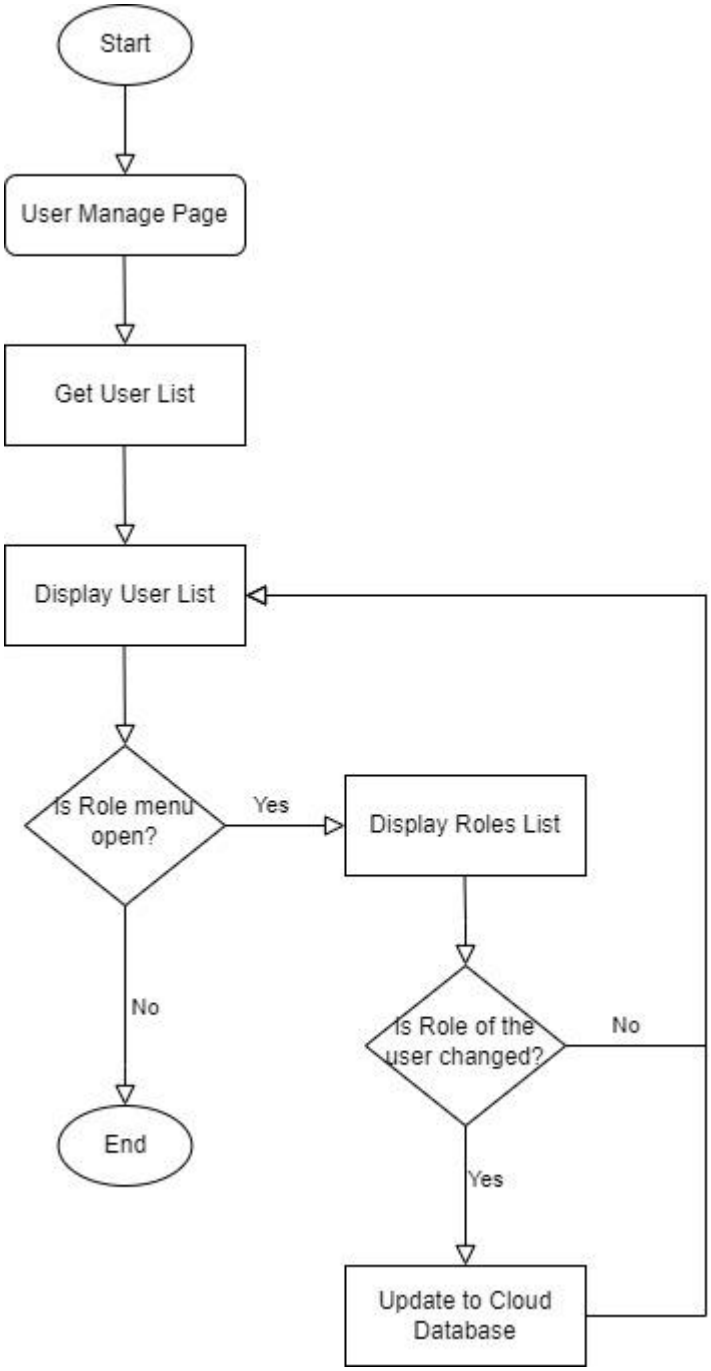


Figure 18 User module flowchart

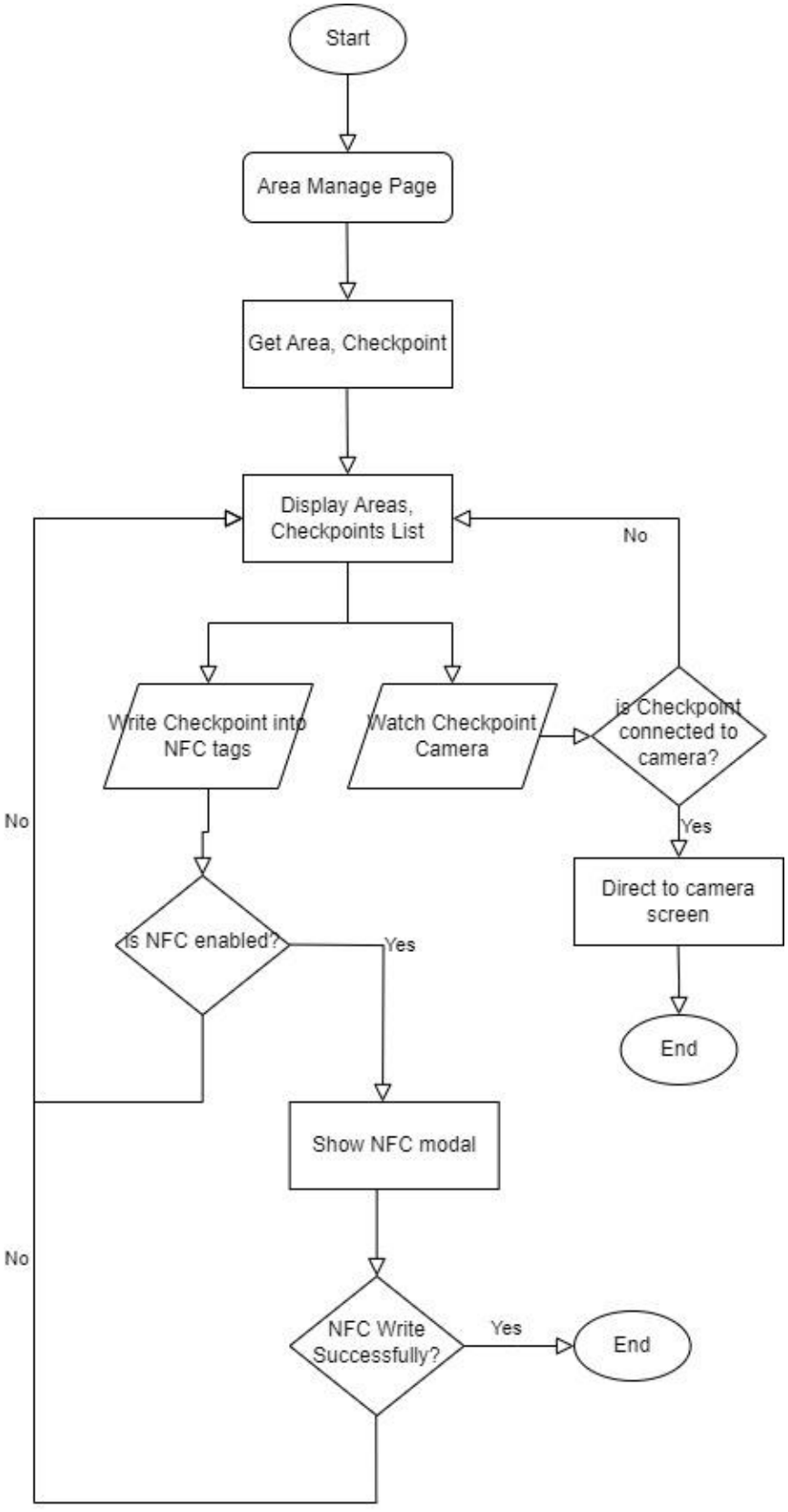


Figure 19 Area module flowchart

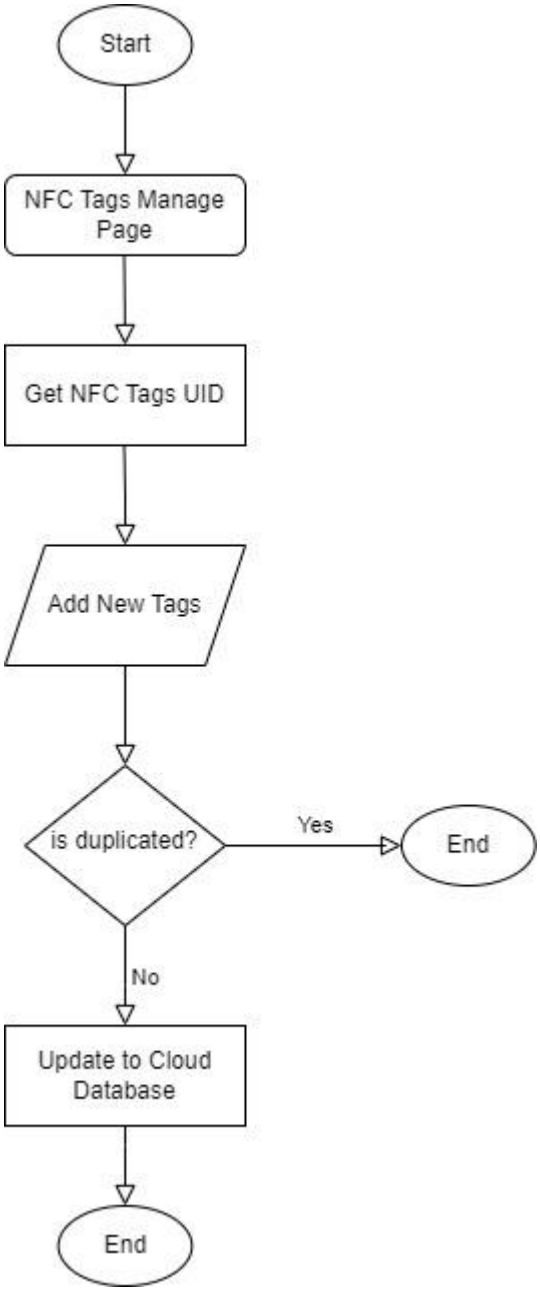


Figure 20 NFC Tags Module flowchart

4.4 System Components Interaction Operations

NFC Scan module flowchart

In the NFC Scan module flowchart, the process begins when the user accesses the Scan page. The system first checks if the device is NFC-enabled. If NFC is not enabled, the functionality ends without proceeding further. If NFC is enabled, a modal pop out prompting the user to scan NFC tags. The system then waits for the user to perform the tag scan.

Once an NFC tag is detected and its data is retrieved, the system attempts to decrypt this information using a secret key. If decryption is not successful, the user is redirected back to the Scan page and close the modal window. If decryption is successful, the system proceeds to verify the NFC tag's UID against records stored in the Cloud Database. If the NFC tag UID is not found in the database, a warning message is displayed and the modal is closed. However, if the UID exists in the database, the system adds a log of the checkpoint and updates the Cloud Database accordingly.

Incident module flowchart

In the Incident module, the process begins when the user navigates to the Incident page. The system allows the user to first select the type of incident that occurred in specific areas. Next, the user can choose to capture a new image or choose an existing image from their device. Finally, the user selects a checkpoint from their assignment’s patrolling area to create an incident record, then update in the Cloud Database. This sequence enables users to efficiently document and report incidents within the system.

Location module flowchart

In the Location module, the process begins when the user navigate to the Location page. The system first checks if it has permission to access the device's location. If permission is not granted, the system requests permission from the user. If permission is still not granted, the process ends. Once permission to access the location is granted, the module displays a Google Map. The system get the latitude and longitude coordinates of checkpoints to point out both the checkpoints and the user's current location on the map.

Timeline module flowchart

In the Timeline module, users can navigate between two tabs which are Pending and History. In the Pending tab, the system retrieves the user's assignment list from the database, which includes the current checkpoint and the scheduled time for the next checkpoint. Using the current checkpoint, the system finds and displays information about the next checkpoint to the user. In the History tab, the system retrieves the areas assigned to the user along with the associated checkpoints and logs from the database. It then displays areas and checkpoints. The logs are displayed in a modal to enhance user experience and facilitate easy access to scanned record.

User module flowchart

In the User module, the process begins when the admin navigates to the User page. The system will get the user list first then display it to the admin. If the user did not open the role menu to change a role of a user, the process end. If the user opens the menu and change the user role, it will then update to the Cloud Database and the authority of the user will be changed between “user” and “admin”. These 2 roles can access to different functionalities of the system.

Area module flowchart

In the Area module, the process begins when the admin navigates to the Area page. The system retrieves area list with their associated checkpoints and displays this information to the admin. The administrator can perform two actions within this module. The first action involves writing checkpointId and areaId data into an NFC tag. The system first checks if the device has NFC capability. If NFC is not enabled, the process returns to the display page. If NFC is enabled, a modal appears, awaiting the admin to scan an NFC tag and write the data onto it. The second action allows the administrator to check cameras installed at checkpoints. Before accessing the camera screen, the system checks if the selected checkpoint has a camera installed. If a camera is there, the system directs the admin to the camera screen. If no camera is installed, the process returns to the display page.

NFC Tags module flowchart

In the NFC Tags module, the process begins when the admin navigates to the NFC tags page. The system gets all the NFC tags UID stored in the database. Then,

admin can add new tags UID into it by just scanning their NFC tags on the device same as the flow we talked just now and the system will check if the UID is duplicated. If it is duplicated, then the process is end. If it is not duplicated, it will update the Cloud Database.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 Hardware Setup

The hardware used in this project is a Lenovo computer and Android mobile device. The computer is used to develop the application using Visual Studio Code with React Native. The mobile device is used to test the Application developed. Since certain functions is better to use a physical device rather than an emulator. Then, the mobile device is NFC enabled while there are 5 NFC tags is also included in the hardware used. The NFC tags include NFC card, stickers and a round coin.

Description	Specifications
Model	LENOVO IdeaPad Gaming 3 15ARH05
Processor	AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz
Operating System	Windows 11
Graphic	NVIDIA GeForce GTX1650Ti
Memory	16GB DDR4 RAM 3200MHz
Storage	512GB SSD

Table 5.1 Specification of Laptop

Description	Specifications
Model	Xiaomi 9T
Processor	Octa-core Max 2.2GHz
Android version	Android 11 RKQ1.200826.002
Memory	6.00 GB
Storage	64.0 GB

Table 5.2 Specification of Phone

Description	Specifications
Chip	Ntag216 (888 bytes)
Protocol	ISO14443A
Working frequency	13.56 MHZ
Reading and writing distance	1 to 5 cm
Reading and writing time	1 to 2 ms

Table 5.3 Specification of NFC tag

5.2 Software Setup

1. React Native Framework

To create hybrid mobile applications that work across platforms, a versatile framework called React-Native is employed. This framework, developed by Facebook developers in 2015, accelerates development by allowing programmers to use the popular programming language JavaScript ES6. It is now feasible to develop mobile apps for both iOS and Android utilising a single codebase for both native platforms. To support complex projects, React-Native requires additional dependencies such as Redux. [6]

2. React Native NFC Manager

The React Native NFC Manager is a library that gives a unified interface for using NFC capabilities into React Native apps, enabling interaction with NFC features seamlessly across Android and iOS platforms. The functionality of this library includes reading NFC tags, writing NFC tags, handling events and peer-to-peer communication.

3. Drawio

A popular web-based diagramming tool called Draw.io, also referred to as "diagrams.net," enables users to create a wide range of diagrams and visual representations. It is renowned for being simple to use, flexible, and collaborative, making it a useful tool for individuals, groups, and organisations operating in a variety of fields, including software development, project management, system architecture, and more. Drawio will be used to demonstrate the concept of the database like ERD, UML diagram.

4. Figma

Figma has established itself as one of the best prototyping and design tools by providing a platform with powerful design features. Figma's versatility and ease of use make it an useful asset for this project to do easy navigation and event handling. It helps in advanced for the development phase.

5. Yawcam

Yawcam is a Java-based webcam software designed for Windows. The main principles behind Yawcam are simplicity and user-friendliness, while also encompassing all standard webcam software features. The main feature we use this software is for streaming video from webcam over the internet or local network,

enabling live viewing from remote locations. Such that, the administrator can check on the surveillance camera whenever he needed.

6. Firebase Realtime Database

Firebase Realtime Database is a cloud database using NoSQL language. The data is synced among all clients in real-time. This database is available all the time even the application goes offline. All the data is stored as a JSON object, when anything updated in this database, your application will receive update with the newest data in real-time.

5.3 Setting and Configuration

In this project, it is using React Native as its framework. And integrate with nativebase and react navigation. NativeBase is a component library prioritizing accessibility and utility, helping in the creation of uniform user interfaces across Android, iOS, and Web platforms while react navigation is to create the navigation structure in your project.

To setup the environment required, first it needs to install the Visual Studio Code as the source-code editor. Since the extension in the Visual Studio Code can help a lot while developing the project. For example, ES7+ React/Redux/React-Native snippets which is for integration with Prettier, ESLint and Prettier ESLint for formatting the code to look neat on save. Next, it needs to install Android Studio for the emulator and the Android SDK as follow the instruction on the React Native official website docs. Then, it can start initializing a new React Native project by the code below.

```
npx react-native@latest init AwesomeProject
```

After initializing the new project, we can use the below code to install dependencies for NativeBase and React Navigation which is two main libraries it is going to use further.

```
npm install native-base react-native-svg@12.1.1 react-native-safe-area-context@3.3.2
```

```
npm install @react-navigation/native
```

After than that, the environment is finish installing and the development can be started now. First, we need to install all the needed library first which included

```
    "@react-native-community/datetimepicker": "^7.6.3",
    "@react-native-community/geolocation": "^3.2.1",
    "@react-native-firebase/app": "^18.8.0",
    "@react-native-firebase/auth": "^18.8.0",
    "@react-native-firebase/database": "^19.0.0",
    "@react-navigation/bottom-tabs": "^6.5.11",
    "@react-navigation/drawer": "^6.6.6",
    "@react-navigation/material-top-tabs": "^6.6.11",
    "@react-navigation/native": "^6.1.9",
    "@react-navigation/stack": "^6.3.20",
    "firebase": "^10.8.0",
    "firebase-admin": "^12.0.0",
```

```
    "material-ui-community-icons": "^0.15.0",
    "native-base": "^3.4.28",
    "react": "18.2.0",
    "react-native": "0.73.3",
    "react-native-crypto-js": "^1.0.0",
    "react-native-geolocation-service": "^5.3.1",
    "react-native-gesture-handler": "^2.14.1",
    "react-native-image-picker": "^7.1.2",
    "react-native-maps": "^1.13.0",
    "react-native-nfc-manager": "^3.14.12",
    "react-native-pager-view": "^6.2.3",
    "react-native-push-notification": "^8.1.1",
    "react-native-safe-area-context": "^3.3.2",
    "react-native-screens": "^3.29.0",
    "react-native-svg": "^12.1.1",
    "react-native-tab-view": "^3.5.2",
    "react-native-vector-icons": "^10.0.3",
```

The provided code shows the project dependencies along with their versions. To install these dependencies, you can use the npm command followed by the dependency name. For example, to install "react-native-nfc-manager", you would run "npm install react-native-nfc-manager".

In the dependencies shown, datetimepicker is included for user convenience when creating new records involving date and time. Geolocation is utilized to retrieve the user's current latitude and longitude coordinates and point them out on the Google Maps API. The firebase/app, auth, and database dependencies are used to connect to the Cloud Firebase Realtime Database for data storage and authentication. Additionally, react-navigation/bottom-tabs, drawer, material-top-tabs, native, and stack are employed to enhance the user experience and navigate between pages. React-native-crypto-js is used for encrypting and decrypting checkpointId and areaId data when interacting with NFC tags. React-native-image-picker enables users to upload or capture images for incident reporting within the Incident page. The react-native-maps dependency integrates with the Google Maps API, displaying maps within the application. Finally, react-native-nfc-manager plays a crucial role in reading and writing data on NFC tags, allowing interaction with NFC-enabled devices and tags within the application. These dependencies collectively contribute to the functionality, security, and user experience of the application, enabling features like

data handling, navigation, location services, and integration with external services such as Google Maps and Firebase Realtime Database.

5.4 System Operation

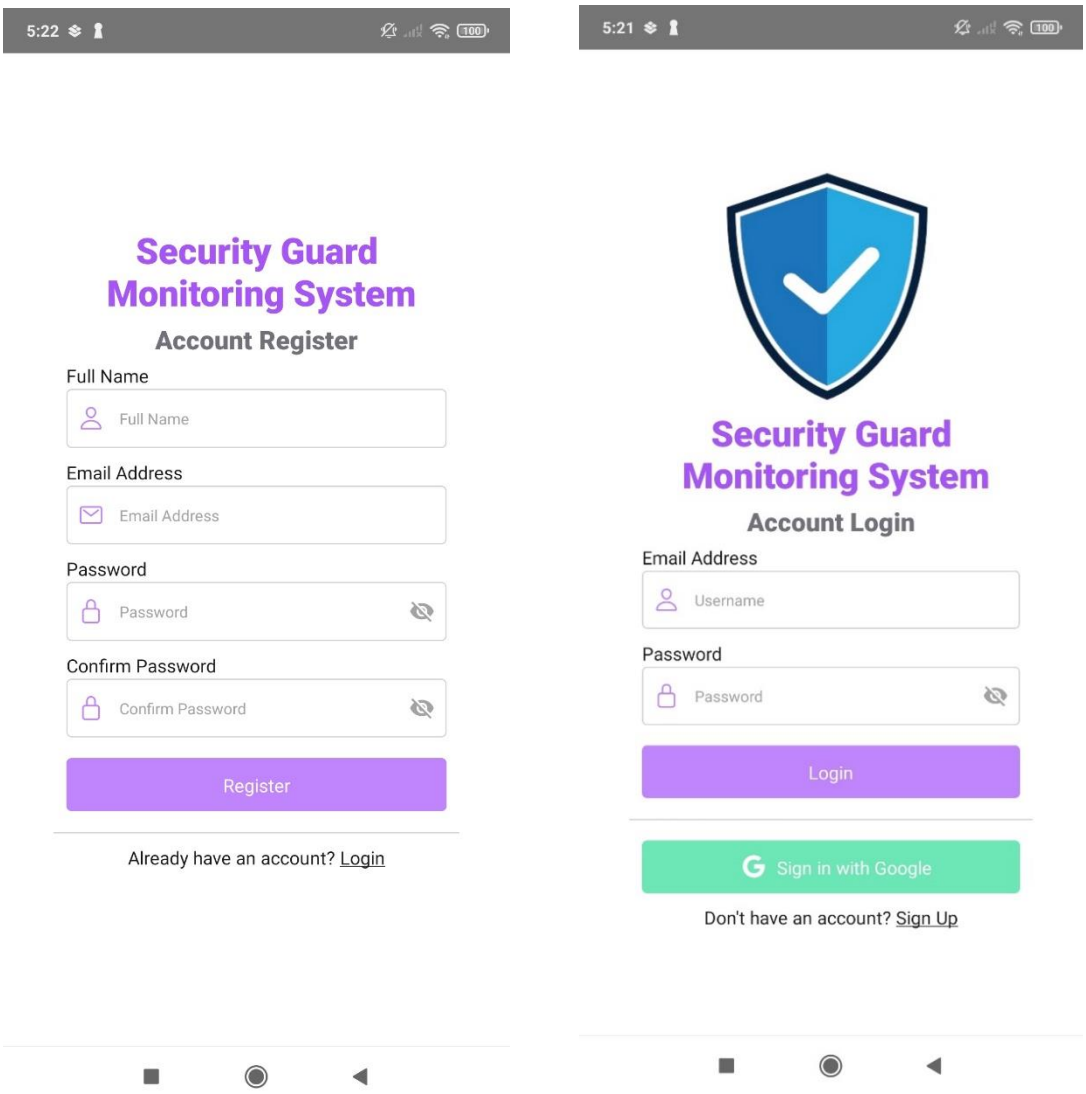


Figure 21 Registration and Login Page

The initial pages users will see are Registration and Login page. New users must register by providing personal information if they're using the application for the first time. Upon registration, users are assigned the default role of "user." When logging in, the system verifies the user's role and navigate them to the appropriate UI page based on their role—either the Admin page for "admin" users or the User home page for "user" users. All the authentication process is using Firebase Authentication.

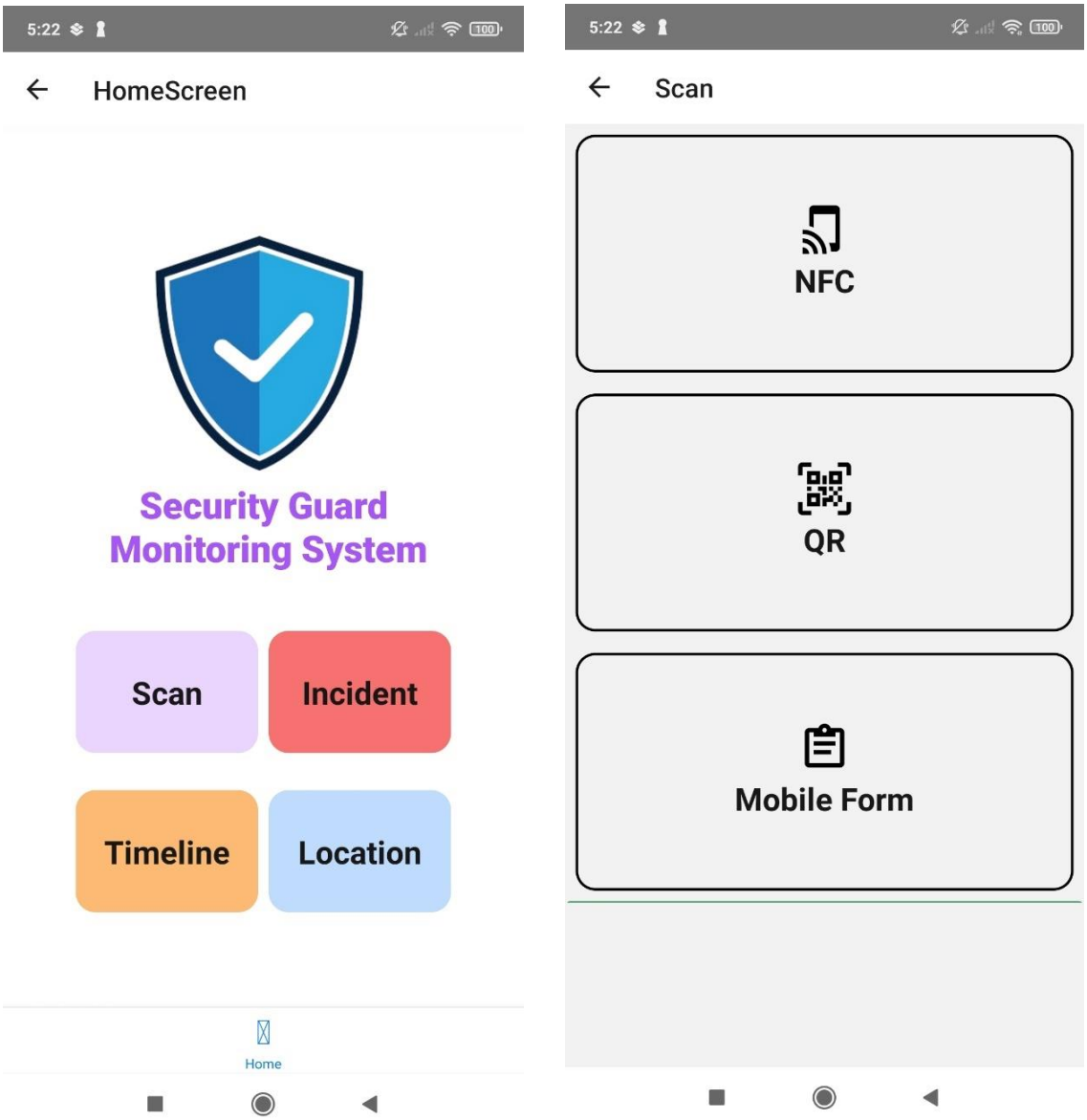


Figure 22 Security Guard Home Screen and NFC Page

On the User home page, users are shown with four buttons that navigate to different pages: "Scan" for taking attendance, "Incident" for uploading multimedia and reporting incidents, "Timeline" for viewing pending assignments and history logs, and "Location" for checking current user location and checkpoint locations.

In the Scan page, there are three attendance options available, but the project specifically focuses on using NFC technology for attendance to ensure data integrity.

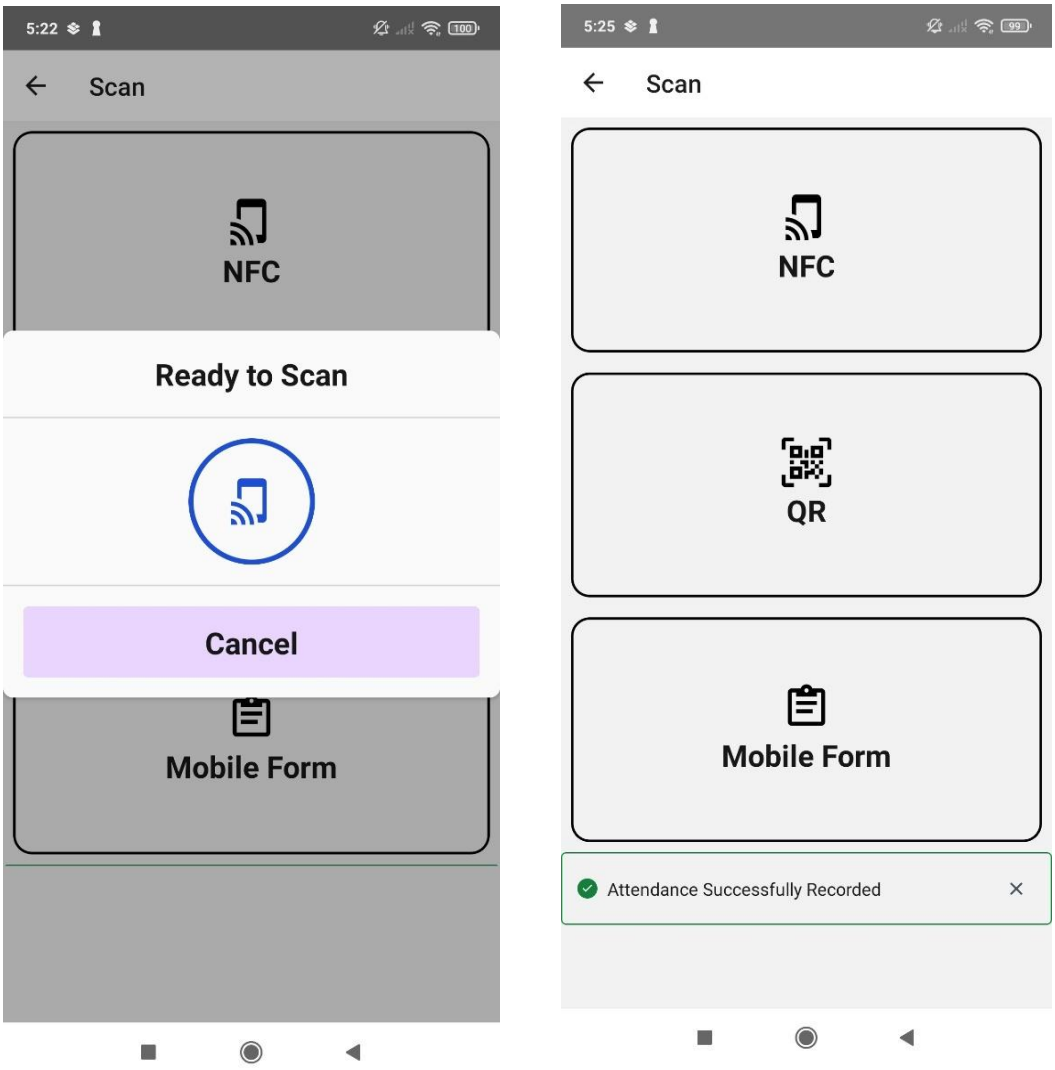


Figure 23 NFC Page

When the user clicks on the NFC option, a modal window will appear indicating that the system is ready to detect NFC tags. The system remains in a waiting state for NFC tags to be scanned while the modal is open, otherwise the system will not detect NFC tag signals. Upon opening the modal and detecting an NFC tag, the system decrypts the data from the NFC tag and checks if the NFC UID exists in the database. If all conditions are met, the system updates the logs in the database, and a dropdown message will be shown to the user that attendance has been successfully recorded. Besides, the system also updates the assignment of the user to record down the user current checkpoint and next scheduled time.

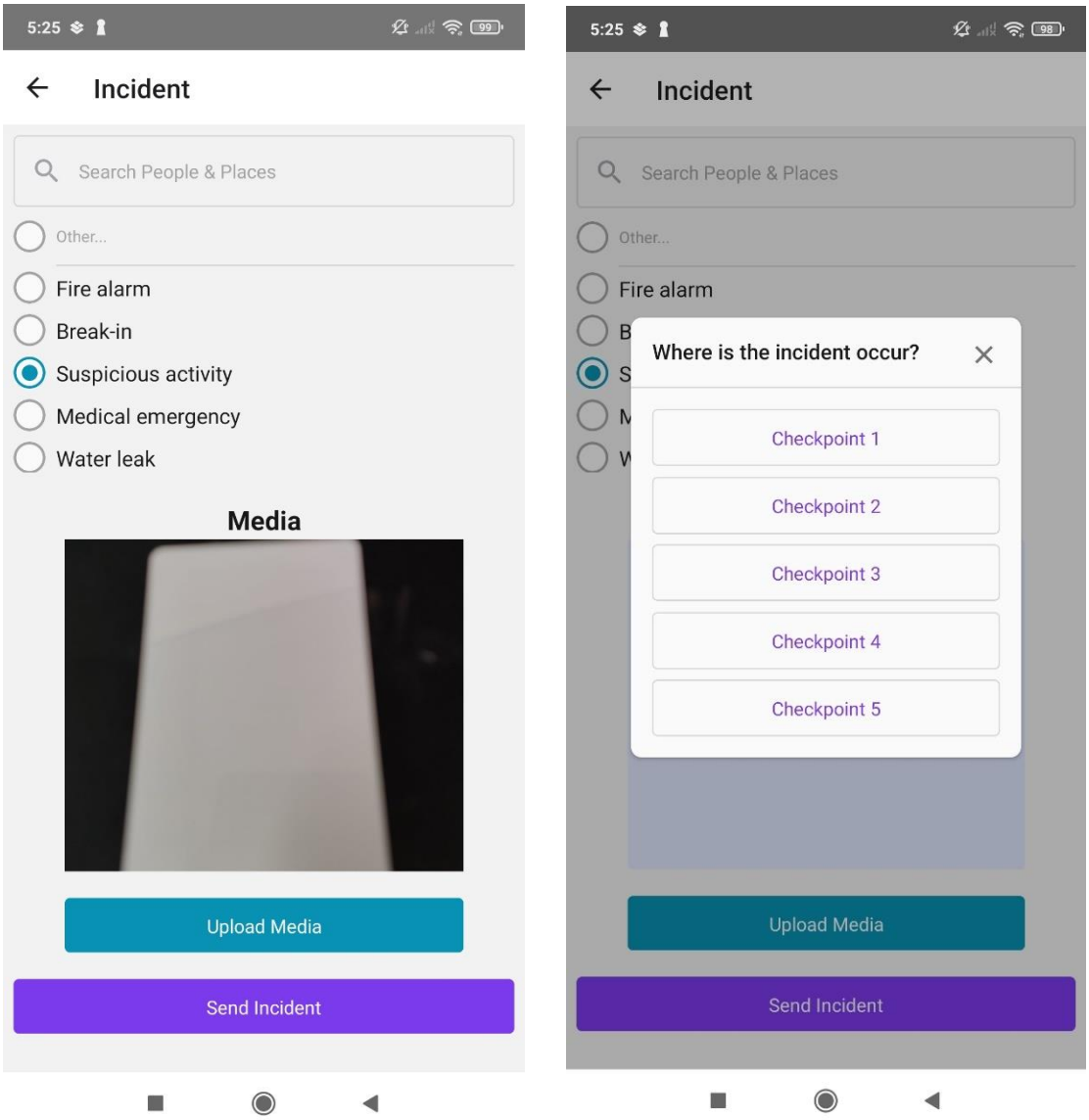


Figure 24 Incident Page

Moving to the Incident page, the user starts by selecting a predefined incident or defining a custom incident. Then, the user can upload an image by either capturing a new photo by clicking on the camera icon under the “Media” or selecting an image from the device by clicking the Upload Media button. Once all necessary information is prepared, the user can submit the incident. A modal window will appear, allowing the user to choose the checkpoints where the incident occurred. Finally, the system updates the database with the incident details.

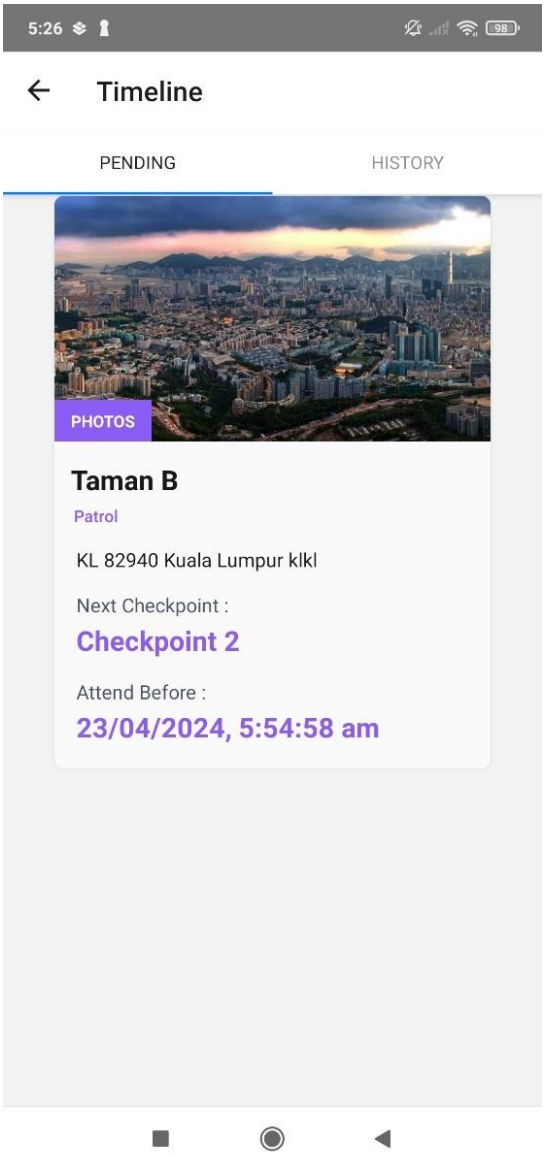


Figure 25 Timeline Pending tab

In the Timeline page, there are two tabs: Pending and History. In the Pending tab, the system retrieves the user's assignments from the database. If the user has attended at least once (in the Scan page using NFC), the system records the user's current checkpoint and identifies the next checkpoint along with its scheduled time. Once all necessary information is determined, it is displayed to the user as shown in the figure.

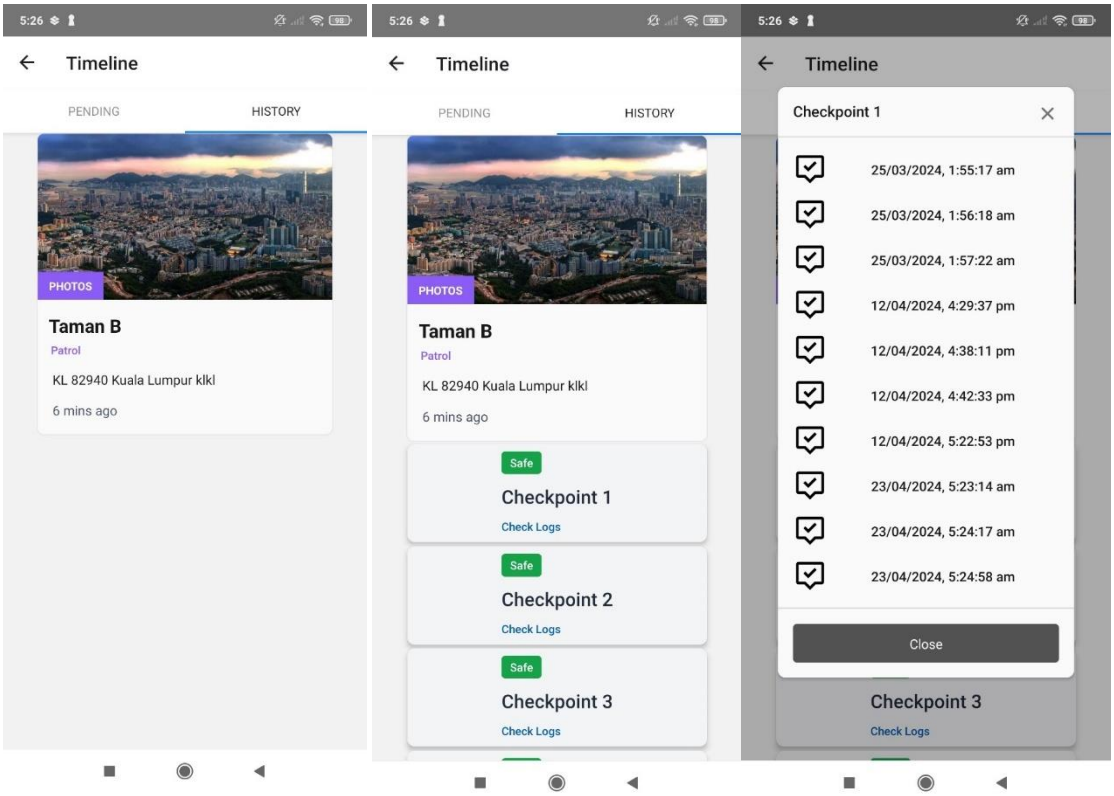


Figure 26 Timeline History tab

Then moving into the History tab, the system retrieves the areas assigned to the user and displays them. The data is structured such that within each area, there are multiple checkpoints, and within each checkpoint, there are multiple logs. This structure is designed to enhance user experience by keeping the data separate and organized. For example, clicking on the "Taman B" image representing an area in the figure will expand to show its checkpoints. Clicking on a checkpoint will trigger a modal displaying all the logs attended by the user for that checkpoint.

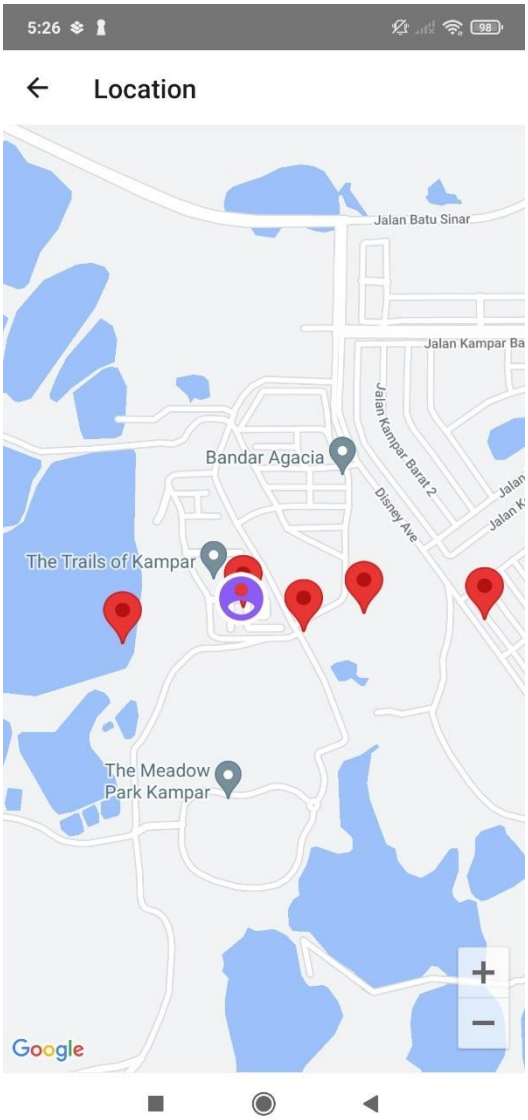


Figure 27 Location page

In the Location page, if the system does not have permission to access the device's location, it will prompt the user to grant permission. Once permission is granted, a Google Map API will be displayed on the page. The map will show a marker indicating the user's current location (purple) and several other markers indicating the checkpoints (red) that the user needs to attend. This feature assists in navigating the user to the checkpoints.

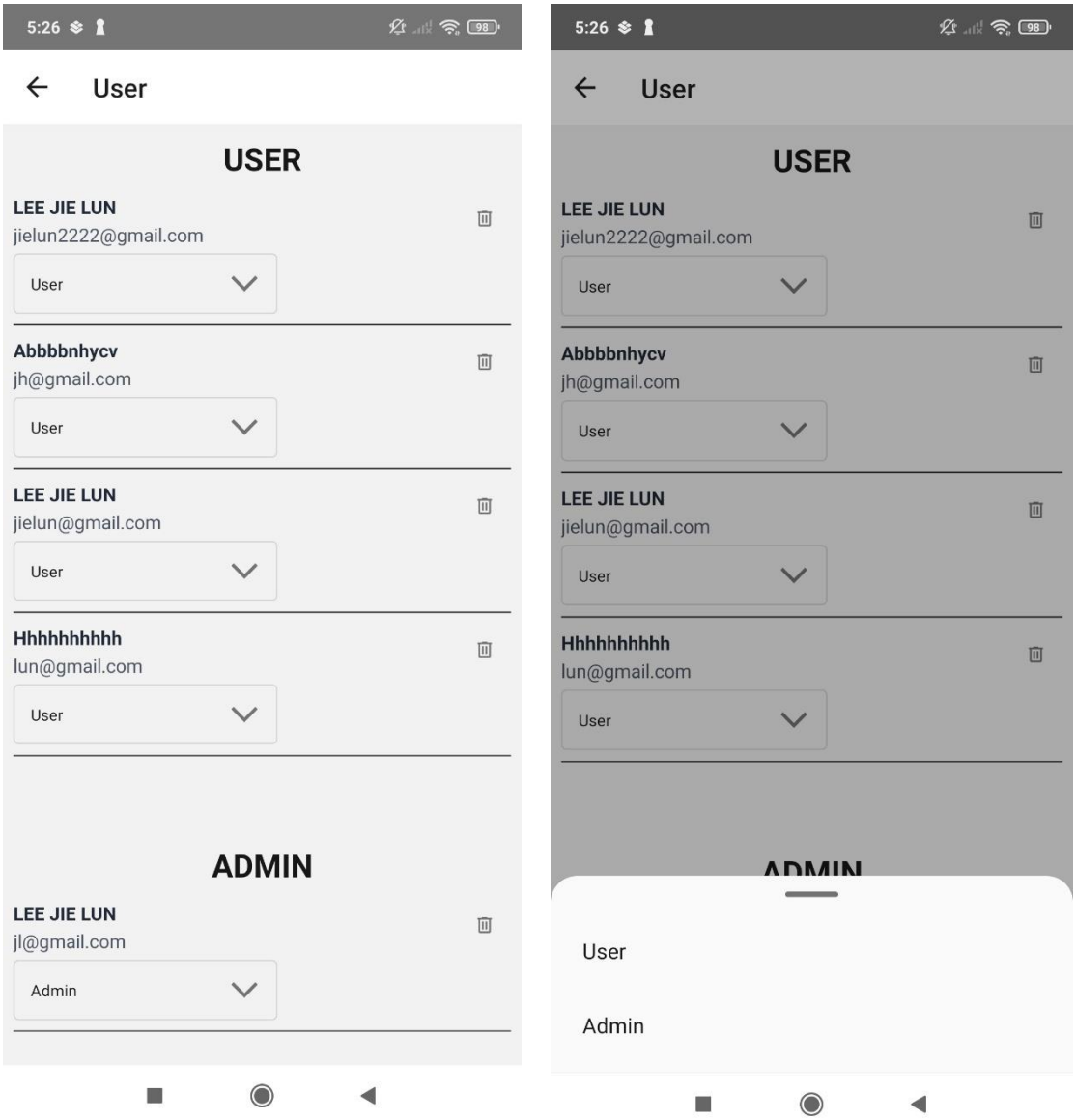


Figure 28 User manage page

Moving into the "admin" role within this system, let's begin with the User page. This page retrieves all user information from the database and presents it to the administrator. The admin has the capability to modify user roles and delete users. By selecting the roles in the dropdown menu, the admin can switch between "user" and "admin" roles. Each role grants access to different functionalities and user interface pages.

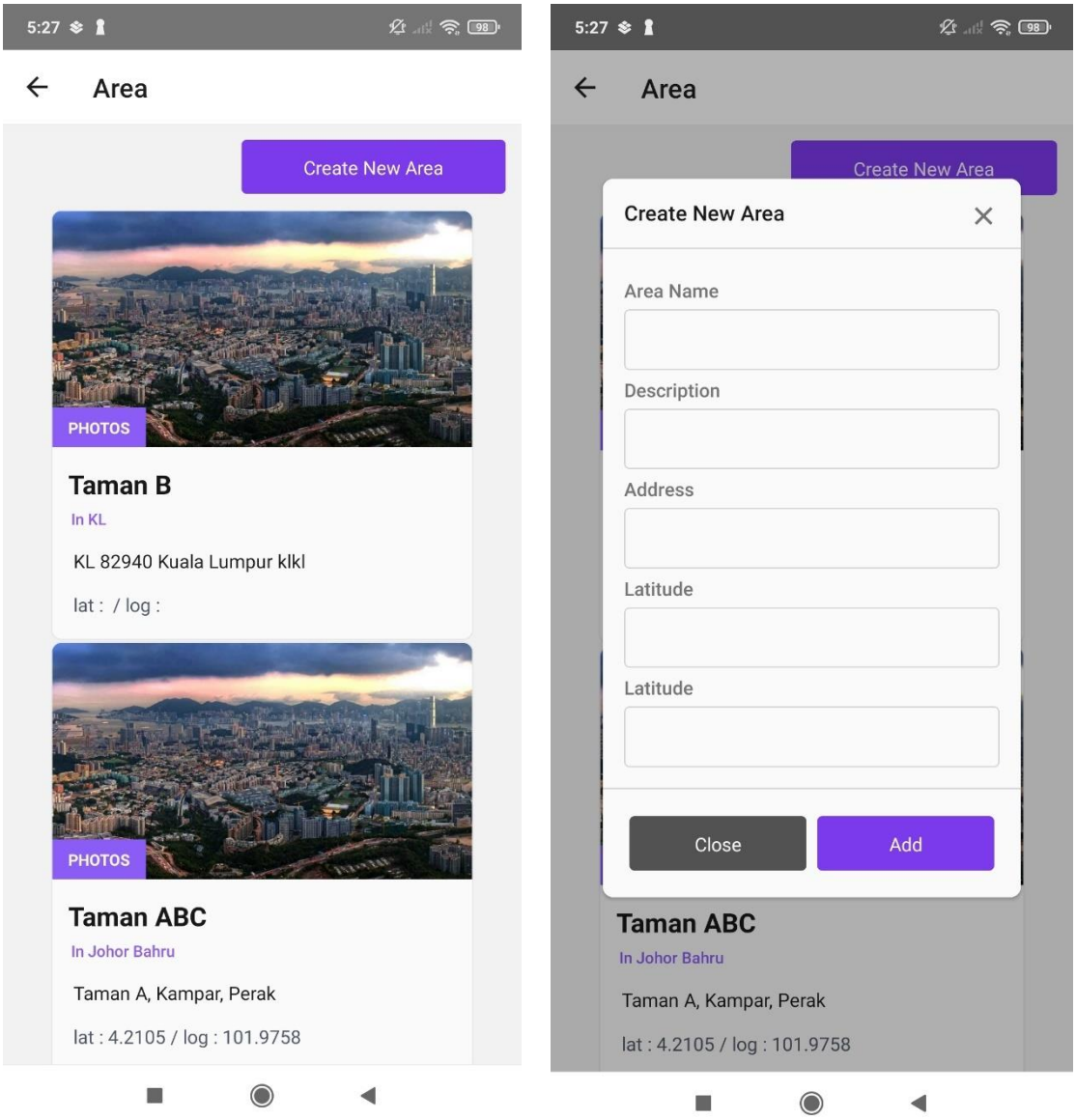


Figure 29 Area manage page

Moving into Area page, the system also get the areas list from the database first and then display to the administrator. But for administrator, he can create new area by providing the required information as shown in the figure.

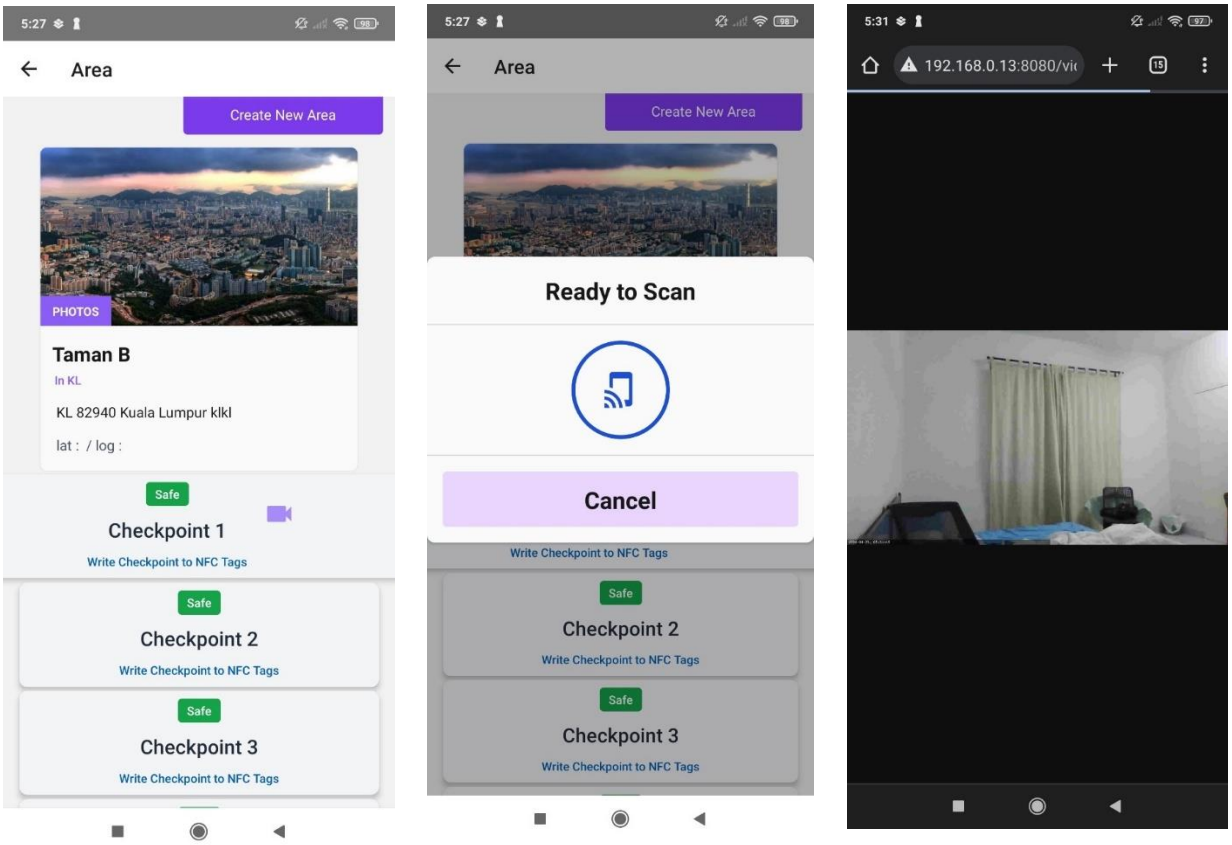


Figure 30 NFC feature and webcam

By pressing the areas image, the checkpoints under the areas will be shown as the figure above. This time pressing the checkpoints does not show the logs, but instead showing the NFC ready to scan modal which mean administrator can setup the NFC tags here by writing the checkpoint information like checkpointId and areaId into the NFC tags here. The system will first encrypt the data and then only write it into the NFC tags. Only the checkpoints with surveillance camera installed will show a video icon beside the checkpoint name as shown in the figure “Checkpoint 1”.

By pressing on the camera icon, it will direct the administrator to a link that connected to the surveillance camera. The link display the video that captured by the camera. This allows administrator to acknowledge current situation of the checkpoint better.

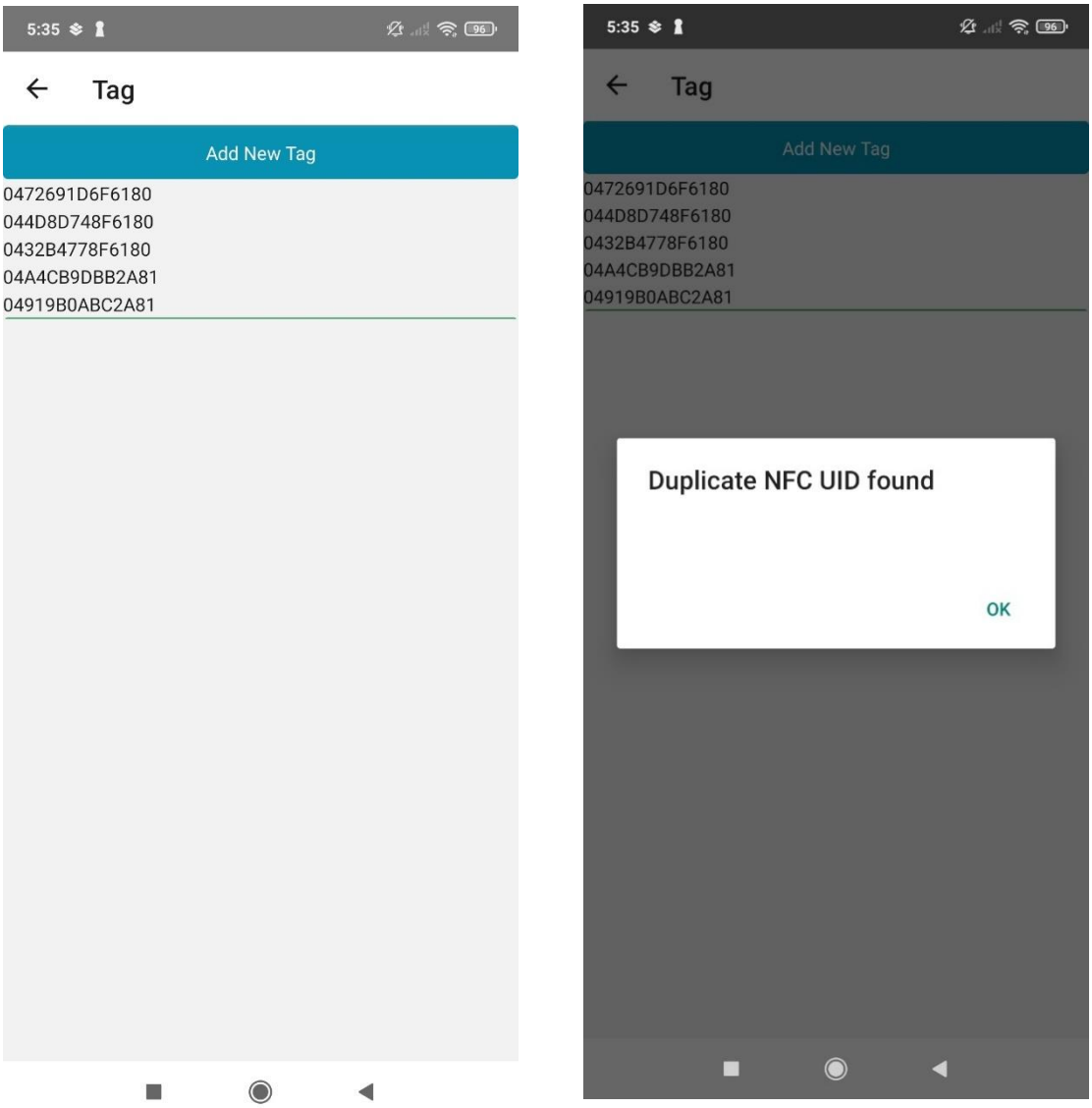


Figure 31 NFC tags page

On the NFC Tags page, the system retrieves the list of NFC UIDs from the database and display them to the administrator. When adding a new tag, a modal window ready for NFC scanning appears, waiting for the administrator to scan their NFC tags. After scanning, the system checks if the NFC tag's UID already exists in the database. If the UID is already exist, a warning message is displayed, as shown in the figure shown.

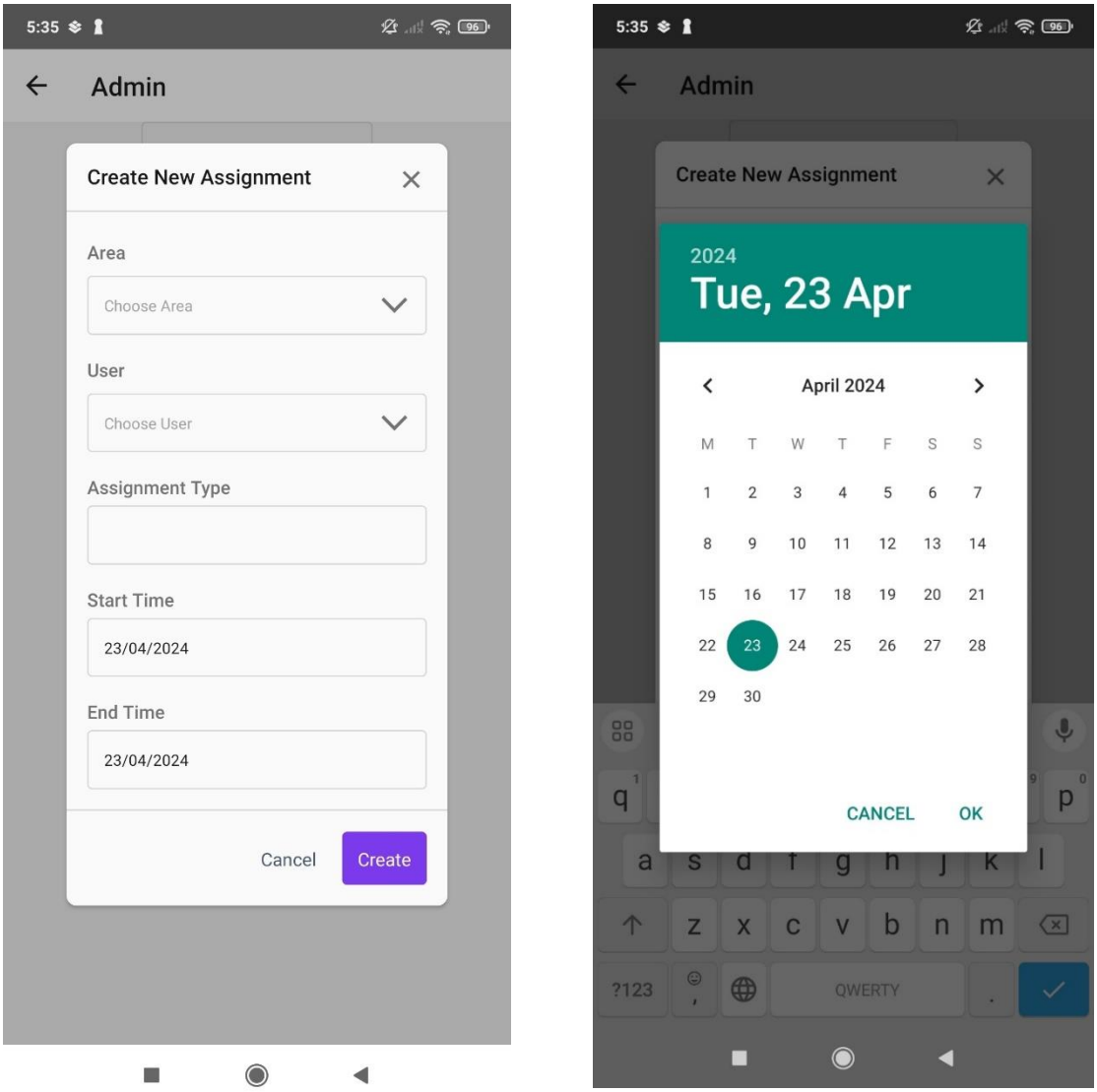


Figure 32 Create assignment page

In Admin home page, admin can create new assignments for the security guard and update the database. The datetime picker is used here.

5.5 Implementation Issues and Challenges

In the implementation of the Security Guard Monitoring System, I encountered specific challenges related to push notification integration and connecting with existing infrastructure. Firstly, I faced difficulties with the integration of Firebase Cloud Messaging for push notifications. Despite efforts to implement scheduled notifications for upcoming checkpoint times, I was unable to achieve successful integration with Firebase Cloud Messaging. This limitation impacted the system's ability to notify users of pending tasks, and real-time communication is unable to achieve.

The Firebase Cloud Messaging feature causes the entire system to crash whenever I try to import the function from firebase/messaging. Although initializing the Firebase app correctly and successfully using other Firebase functions like getAuth from firebase/auth and getDatabase from firebase/database, the getMessaging function encounters an error which is "cannot read property 'addEventListener' of undefined". This issue indicates that there is a specific problem with the getMessaging function, and I will find out and resolve this issue in the future to enable the implementation of this useful functionality.

```
const firebase = initializeApp(firebaseConfig);
const messaging = getMessaging(firebase);
```

Figure 33 Firebase Cloud Messaging Error

Secondly, while I successfully integrated the system with Yawcam to connect to surveillance cameras (using computer webcam as demo), a significant challenge emerged regarding access limitations. The implementation was effective within a private network environment, enabling administrators to remotely monitor checkpoints. However, the system failed to support public access, restricting surveillance camera access to within the same network. This limitation leads to practical challenges, particularly when administrators were not physically present in the same location as security guards, limiting their ability to monitor surveillance feeds remotely.

I have tried to configure my router by setting up port forwarding and making adjustments to allow access via the public IP, but I was unsuccessful in streaming my webcam on the public IP. This failure may be attributed to my limited understanding of networking concepts. Moving forward, I plan to enhance my knowledge in this area to improve the system and address this issue in the future. Addressing these issues will be crucial to enhance system accessibility and functionality across diverse network environments.

5.6 Concluding Remark

In conclusion of system implementation, it specifies that the hardware setup used in this project which using a laptop for development like coding and testing, an android phone is used as a testing device and 5 NFC tags serve as the NFC tags installed at each checkpoint. For the software used in this project, there are React Native framework, React Native NFC Manager, Figma for prototyping, Draw.io for illustrating diagram, Yawcam to capture and host an IP address to show the video, and using Firebase Realtime Database to serve as the database used.

In addition, this chapter specific the setup and configuration needed to create the environment for this project. It mainly introduces the way of creating a new React Native project associated with important library going to be used in this project. It also outlines the version of each library used in this project which can make it ease when rebuilding this system.

Furthermore, it showcases all the operations and screen appear in this system. Walking through the entire system and explain what operation can be done in each screen. This allows reader has an overall concept on my system before looking into the test case in the next chapter.

Lastly, it shows the challenges and issues that have been met while developing this system. There are 2 main issue that have been met which are failing implementation of push notification with Firebase Cloud Database and implementation of webcam public access. From these challenges, valuable information were gained, highlighting areas for future focus and enhancement.

References

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics

Test Case 1: To test whether user can sign up successfully.

Preconditions: User navigates to the registration page.

Steps:

1. Enter valid registration details (name, email, password).
2. Click the "Register" button.
3. Navigate to Login page
4. Enter the email address and password used in Step 1
5. Click “Login” button.

Expected outcomes: User create an account and the system navigate to security guard home page when login.

Result:

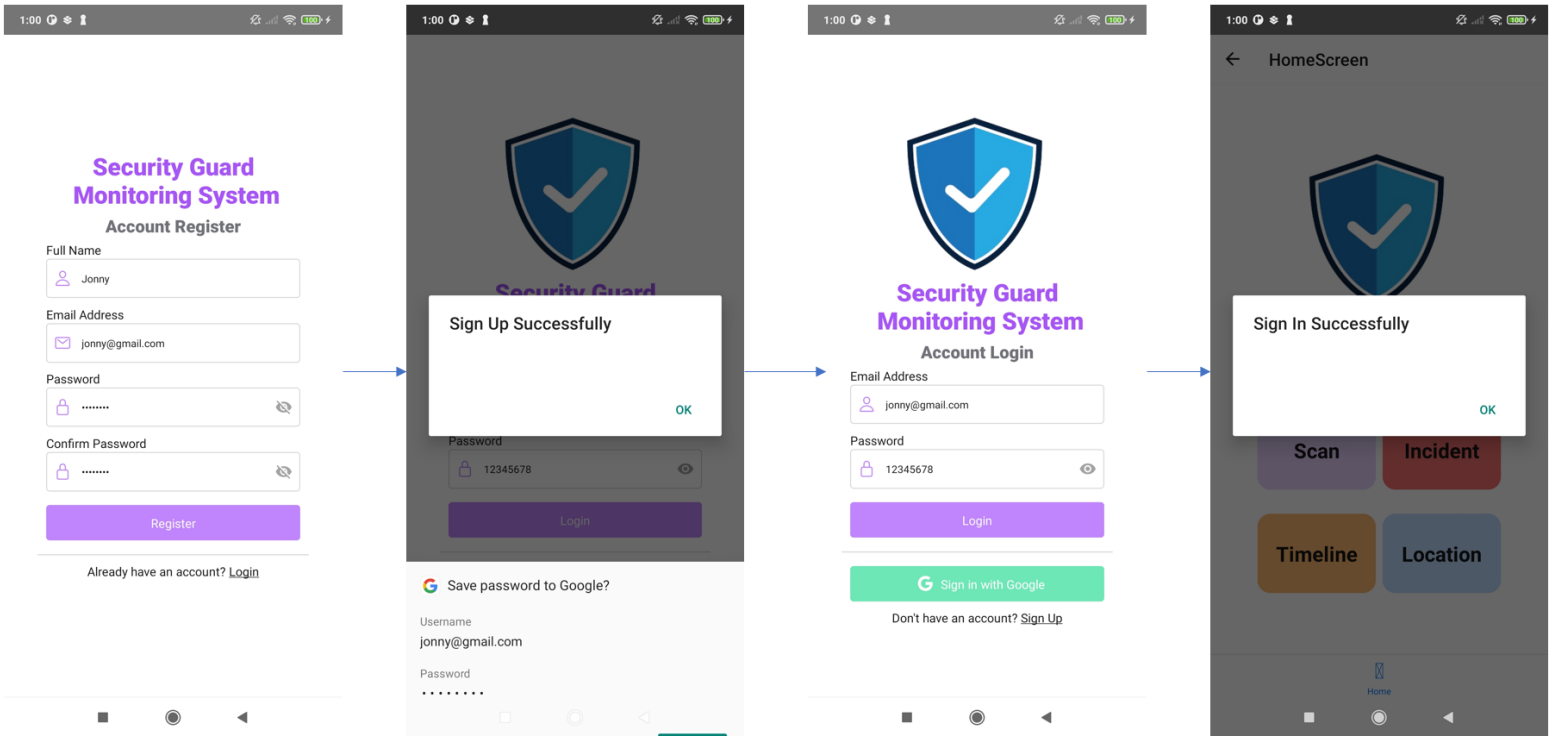


Figure 34 Test Case 1

References

Test Case 2: To test whether admin can sign in successfully to Admin page.

Preconditions: There is an existing “admin” role user in the system with email “jl@gmail.com”

Steps:

1. Enter the email address and password of existing “admin” role user
2. Click “Login” button.

Expected outcomes: Navigate to Admin home page.

Result:

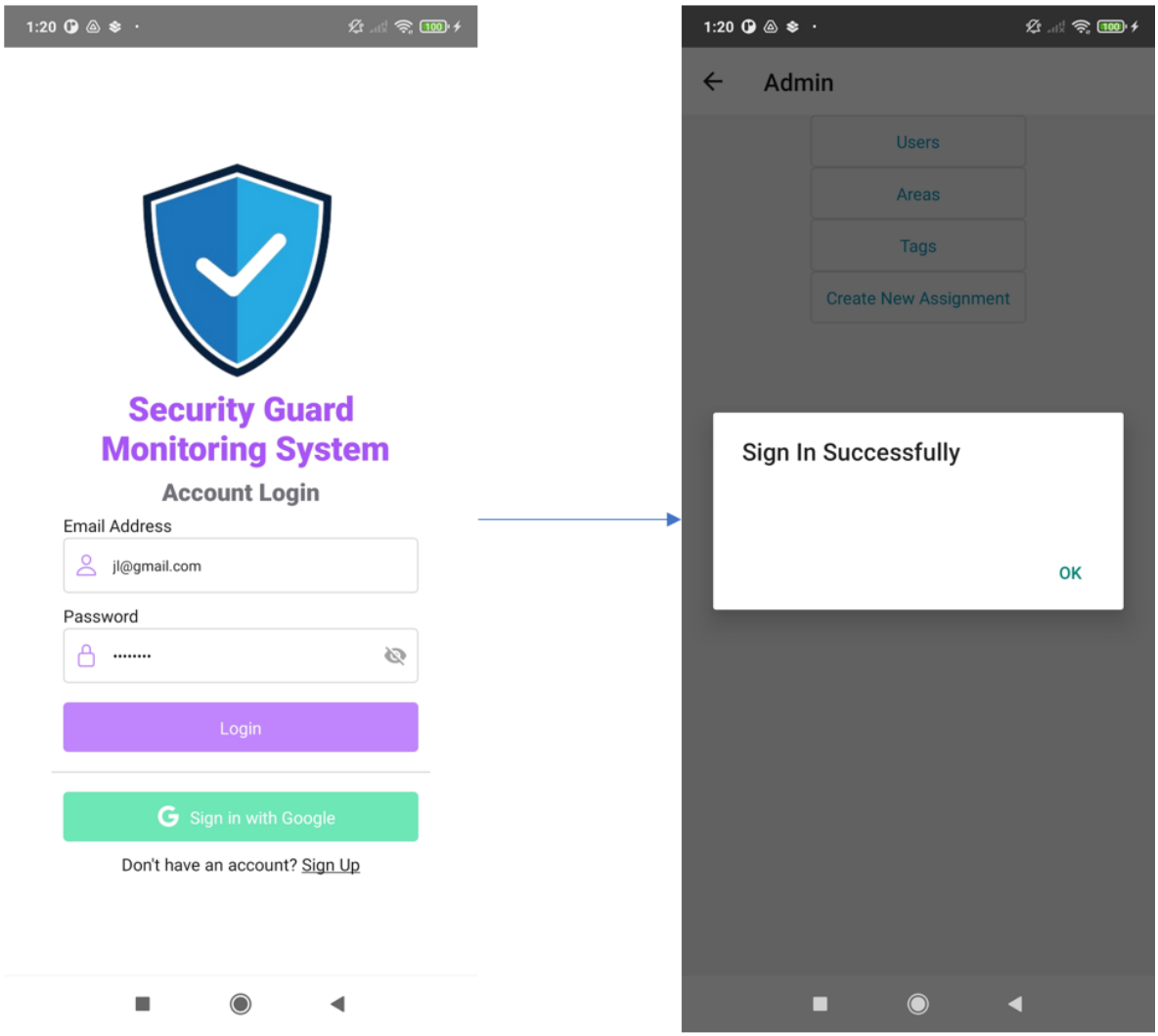


Figure 35 Test Case 2

References

Test Case 3: To test whether user can sign in successfully to User page.

Preconditions: Test case 1 is successful and use the account created in test case 1.

Steps:

1. Enter the email address and password of test case 1 account.
2. Click “Login” button.

Expected outcomes: Navigate to security guard home screen.

Result:

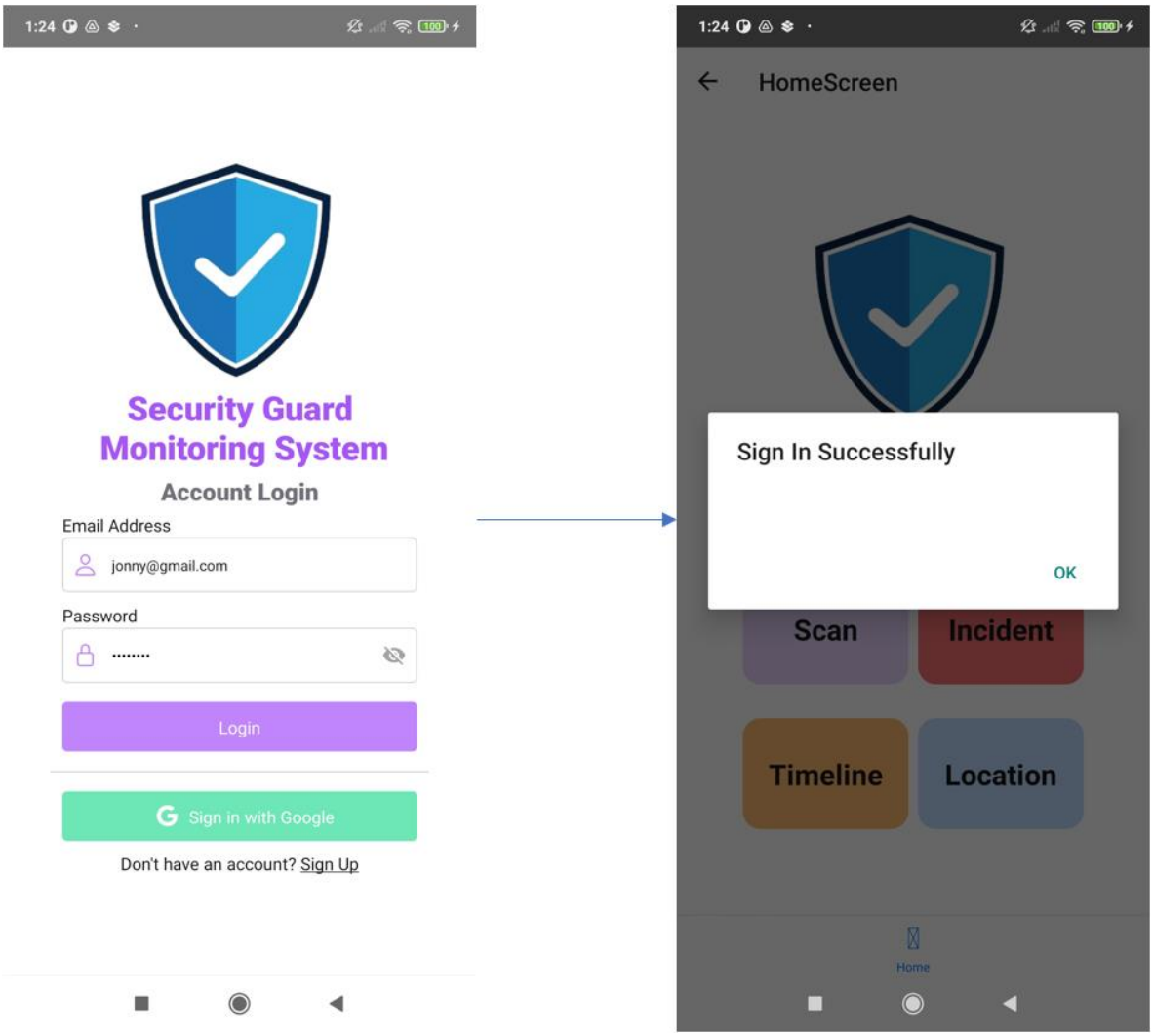


Figure 36 Test Case 3

References

Test Case 4: To test whether admin can change user role from “user” to “admin” successfully.

Preconditions: As an admin, navigate to Admin home page

Steps:

1. Click “Users” button to navigate to User page.
2. Click select component under the user create in Test Case 1 who name is Jonny.
3. Change role of Jonny to “Admin”.

Expected outcomes: Jonny will be directed to Admin home page when logging in.

Result:

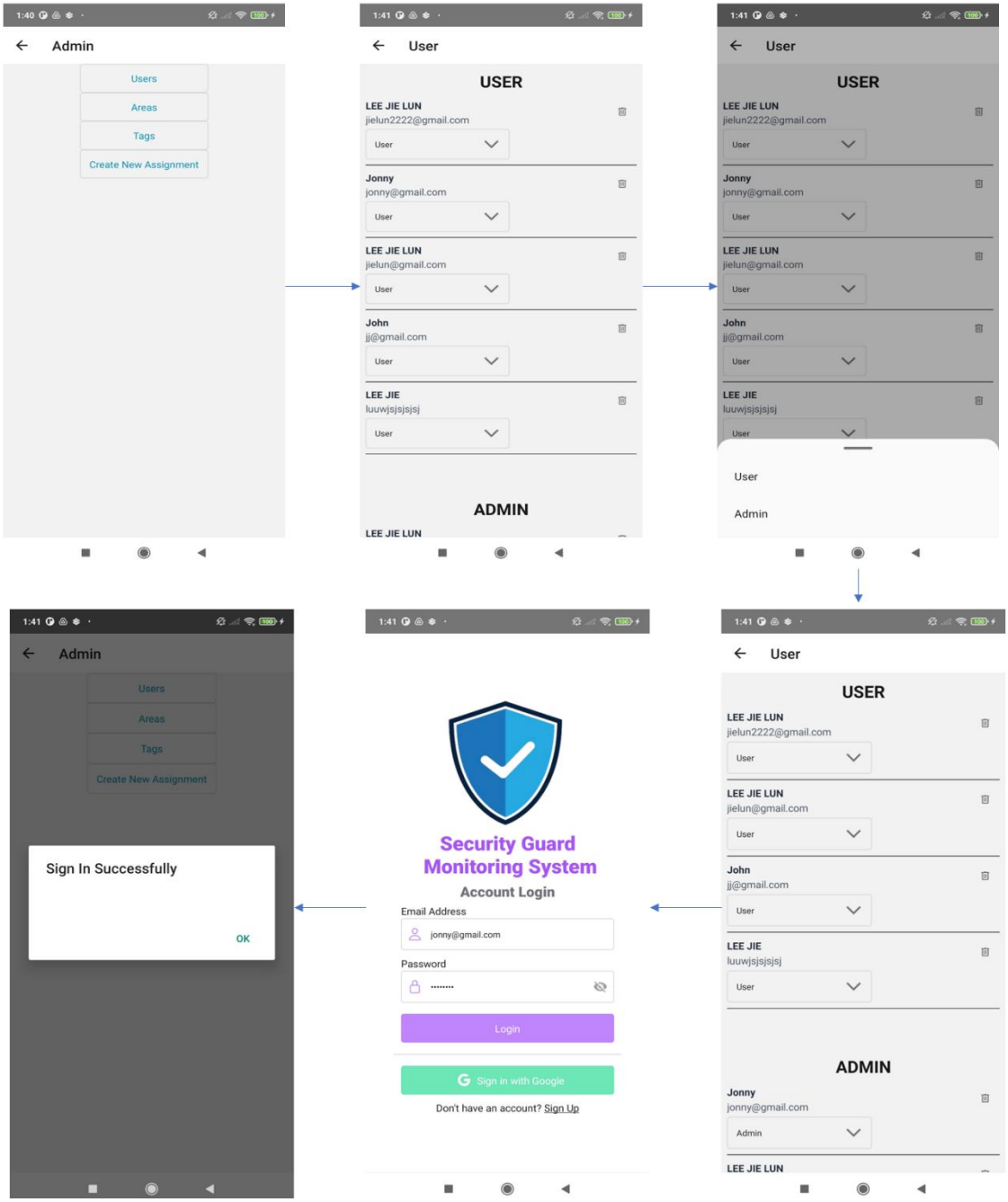


Figure 37 Test Case 4

References

Test Case 5: To test whether admin can create new area successfully.

Preconditions: As an admin, navigate to Admin home page

Steps:

1. Click “Area” button to navigate to Area page.
2. Click “Create New Area” button to open Create New Area modal.
3. Enter the details needed
4. Click “Add” button.

Expected outcomes: New Area is successfully added to the database and display to admin.

Result:

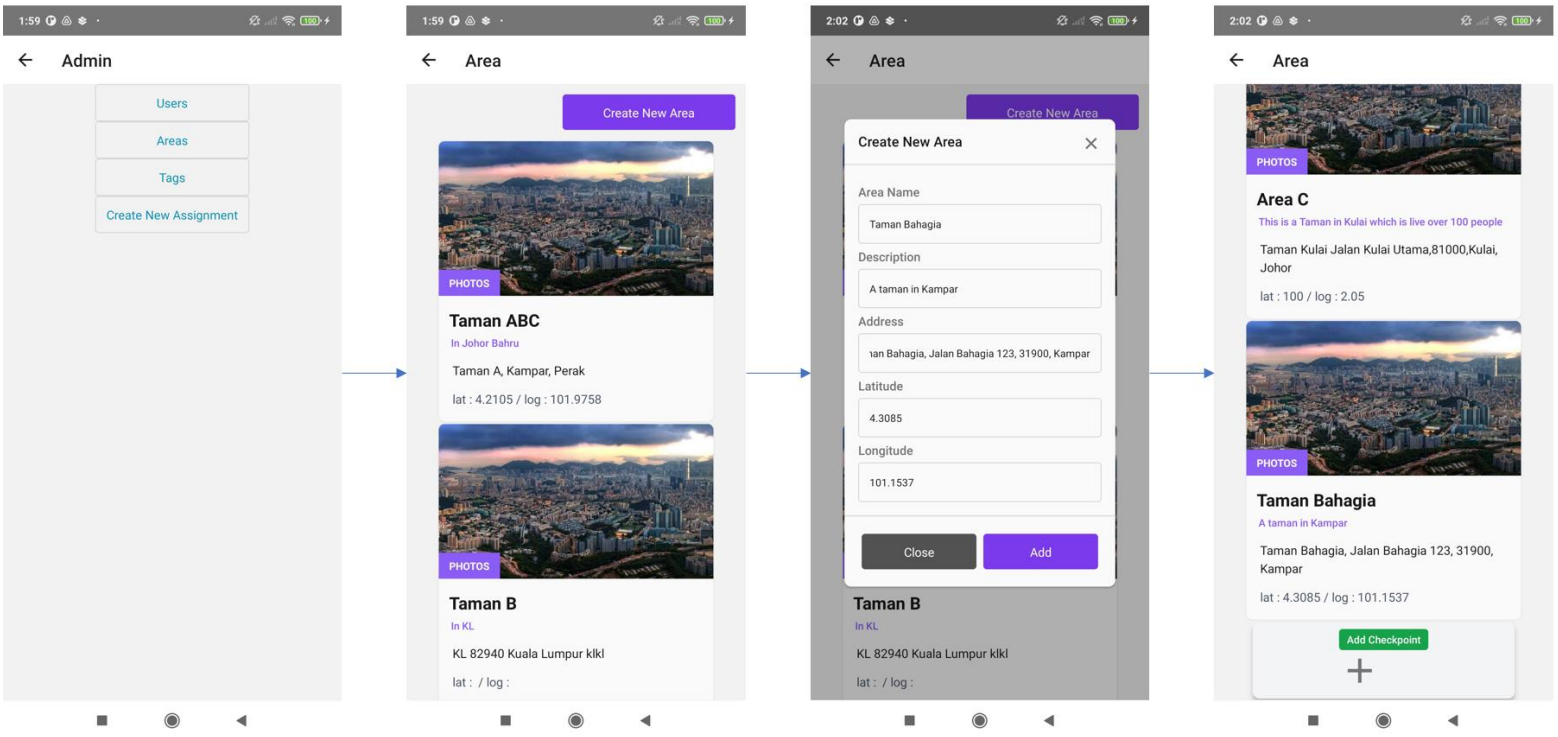


Figure 38 Test Case 5

References

Test Case 6: To test whether admin can create new checkpoint successfully.

Preconditions: As an admin, navigate to Area page

Steps:

1. Click on the Area we create in Test Case 5 and a collapse will dropdown.
2. Click on the “Add Checkpoint” and a modal will pop out.
3. Enter required details.
4. Click “Add” button.

Expected outcomes: New Checkpoint is successfully added under the area created in Test Case 5 to the database and display to admin.

Result:

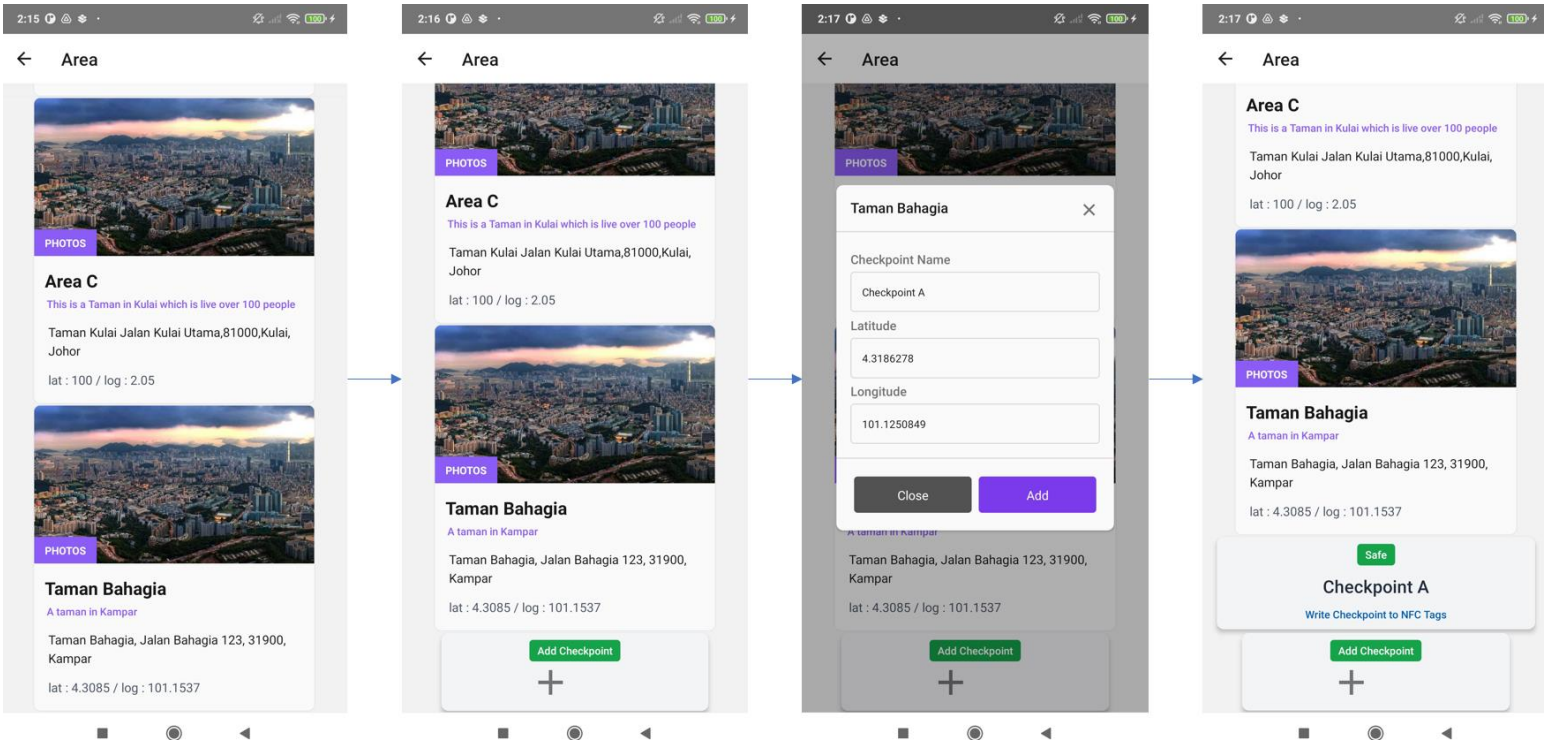


Figure 39 Test Case 6

References

Test Case 7: To test whether admin can write encrypted checkpointId and areaId into NFC tags successfully.

Preconditions: As an admin, navigate to Area page

Steps:

1. Click on the Area we create in Test Case 5 and a collapse will dropdown.
2. Click on the Checkpoint we create in Test Case 6 and a modal will pop out saying it is ready for NFC.
3. Put NFC tags near the device.

Expected outcomes: Encrypted data is stored into the NFC tags.

Result:

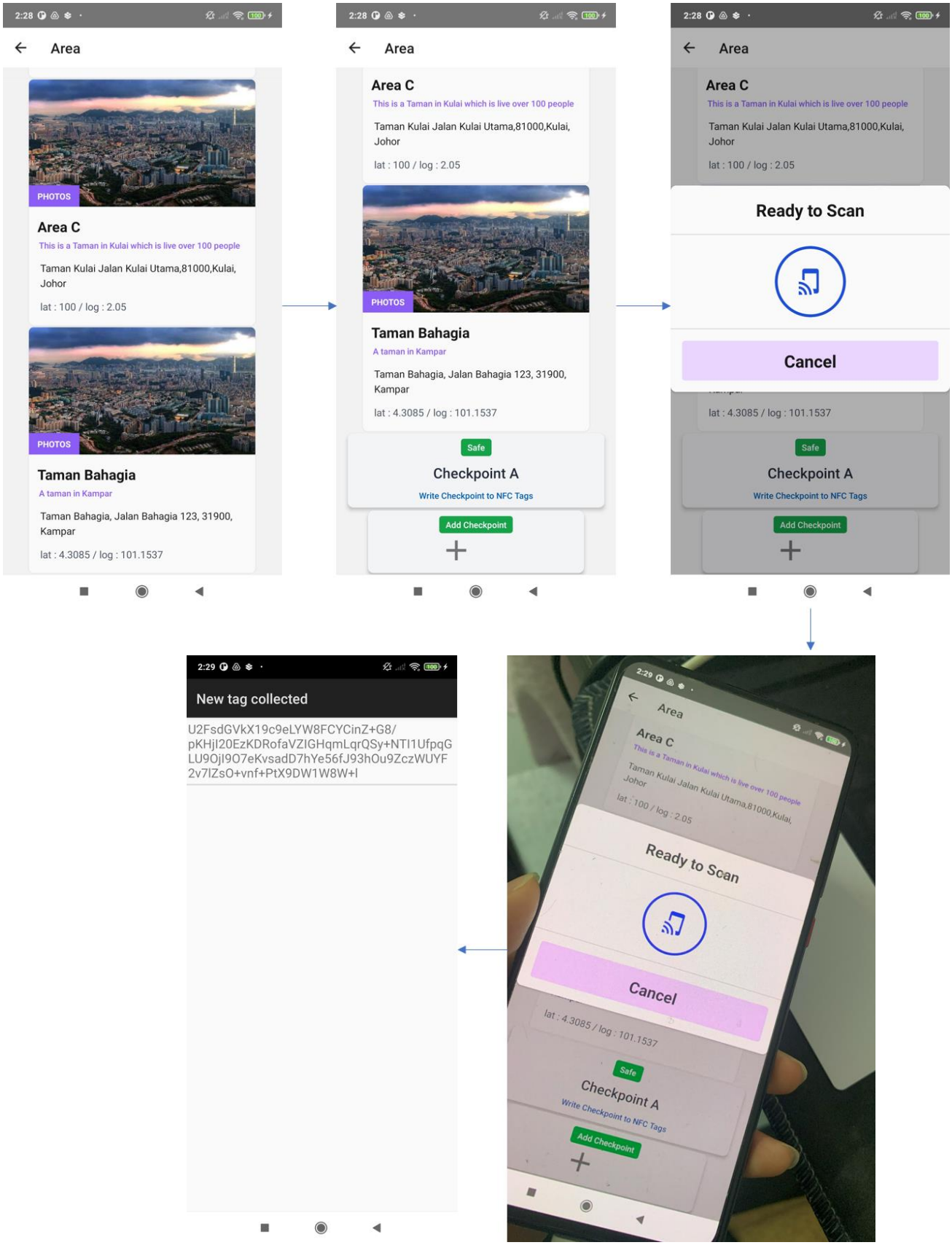


Figure 40 Test Case 7

References

Test Case 8: To test whether admin can check the surveillance camera of checkpoints successfully.

Preconditions: As an admin, navigate to Area page and click on the area created on Test Case 5, and a webcam or camera should integrate with Checkpoint A (store an IP address in checkpoint table)

Steps:

1. Click on the camera icon and it will direct to a browser.
2. Choose a browser you like.

Expected outcomes: The directed IP address show what the webcam is capturing.

Result:

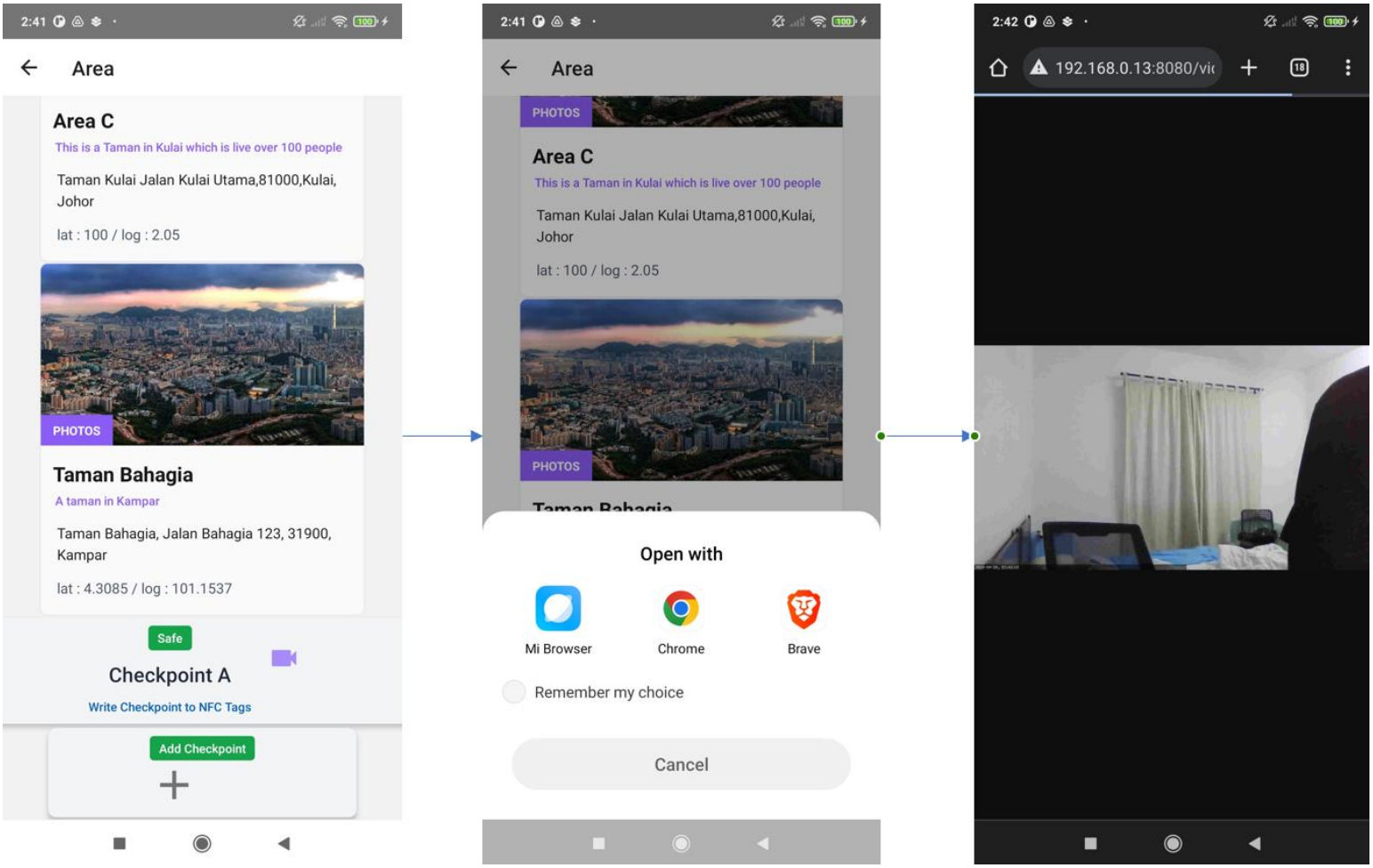


Figure 41 Test Case 8

References

Test Case 9: To test whether admin can add NFC tags UID successfully.

Preconditions: As an admin, navigate to Admin home page.

Steps:

1. Click “Tags” button.
2. Click “Add New Tag” button and a modal will appear.
3. Put the device close to the NFC tag.

Expected outcomes: The NFC UID is updated to database and display to user.

Result:

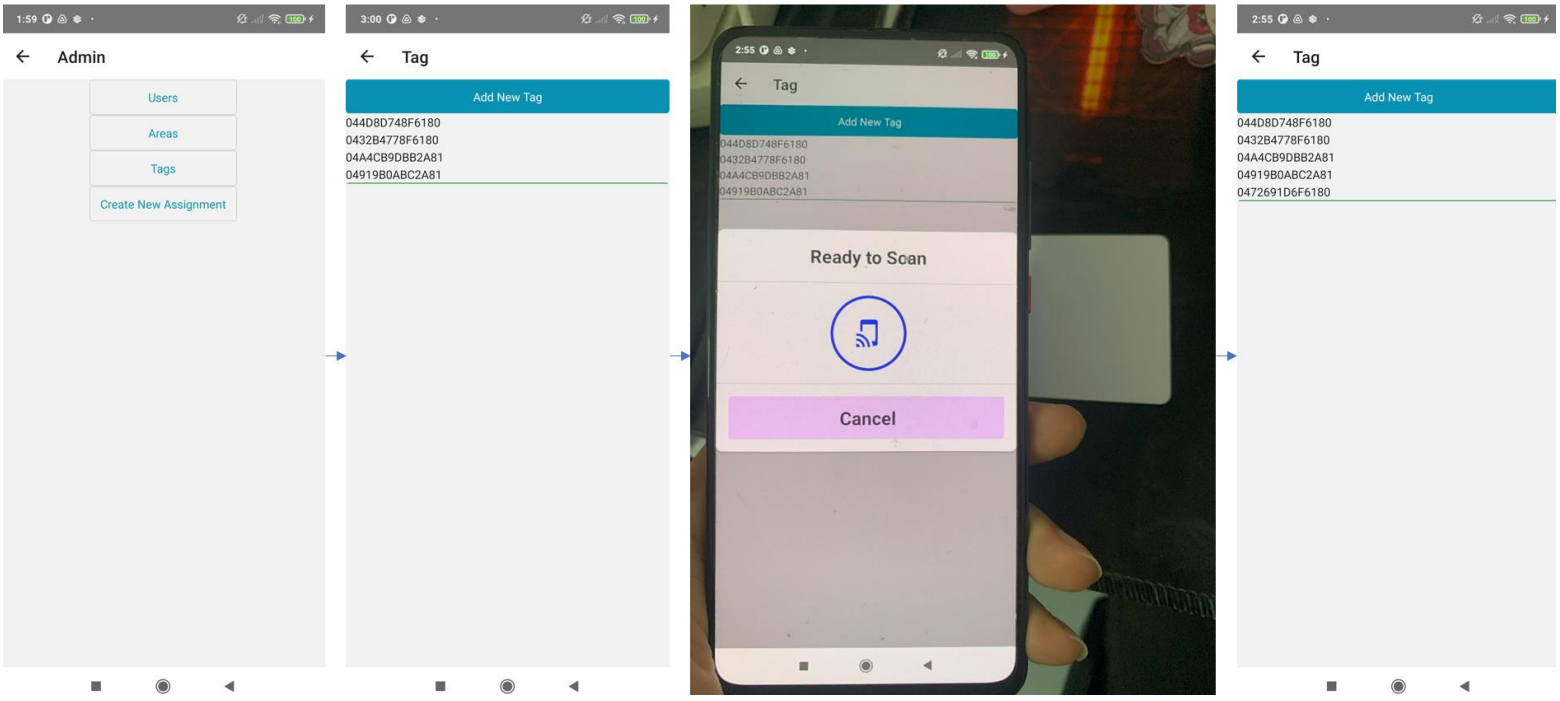


Figure 42 Test Case 9

References

Test Case 10: To test whether admin can create new assignment to a user.

Preconditions: As an admin, change the role of the user create in Test Case 1, Jonny back to user, then navigate to Admin home page.

Steps:

1. Click “Create New Assignment” button and a modal will appear.
2. Create an assignment for Jonny with Taman Bahagia Checkpoint A.
3. Click “Create” button.
4. Logout.
5. Login using Jonny account and navigate to security guard home page.
6. Click “Timeline” button.

Expected outcomes: The Pending is showing the assignment created in Step 1-3.

Result:

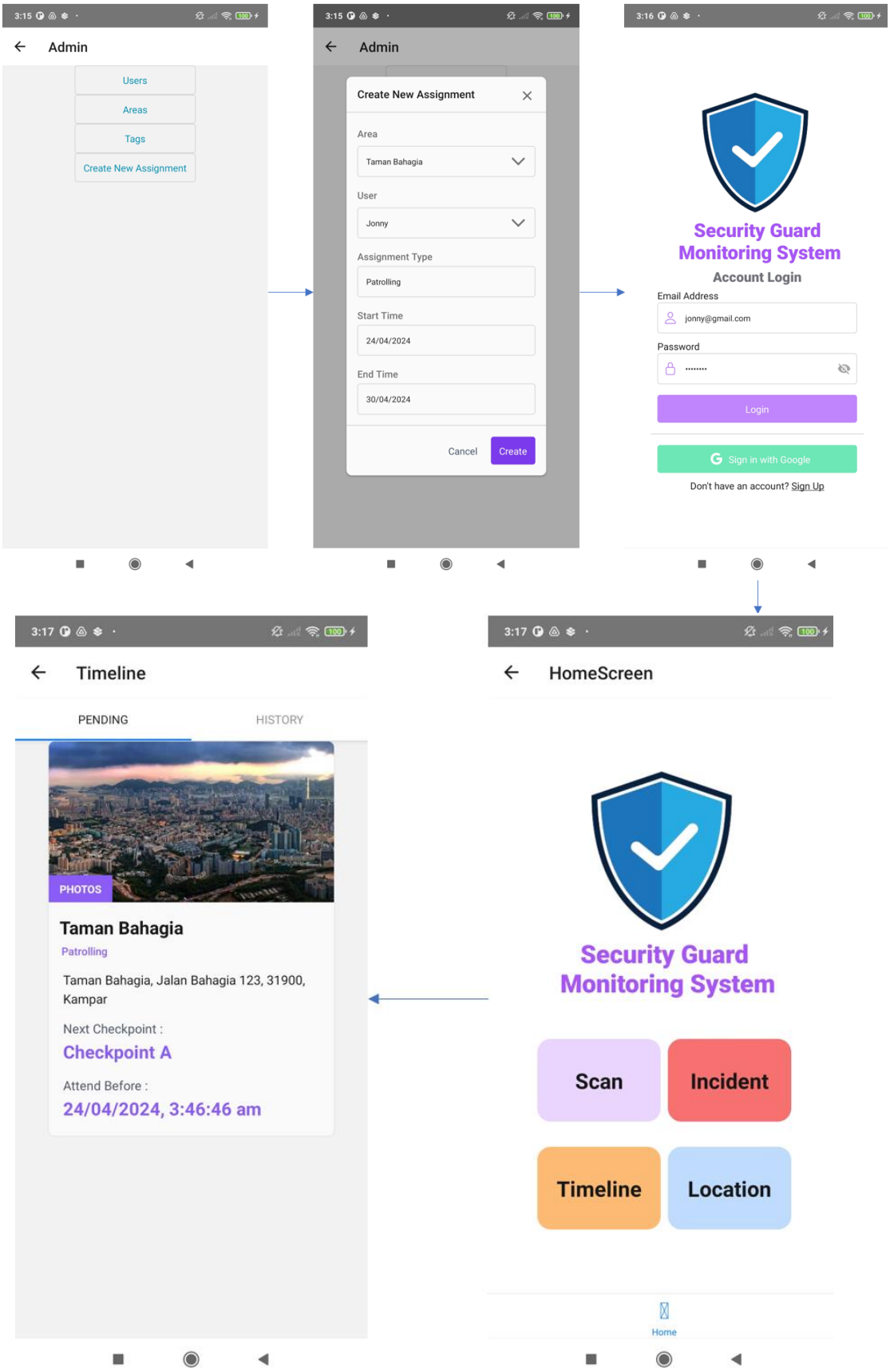


Figure 43 Test Case 10

Test Case 11: To test whether security guard can scan the NFC tags to take attendance successfully.

Preconditions: As an admin, create 2 more checkpoints for Taman Bahagia. Login as a security guard (Jonny we created Test Case 1), navigate to security guard home page

Steps:

1. Click “Scan” button.
2. Click “NFC” button.
3. Put the NFC tag near your device.
4. Go back to HomeScreen and Click “Timeline” button.

Expected outcomes: The system decrypts the data stored in the NFC tags and store the logs into the database. In Timeline page, Pending tab should show “Attend Before” is current time + 30 minutes and the next checkpoint name.

Result:

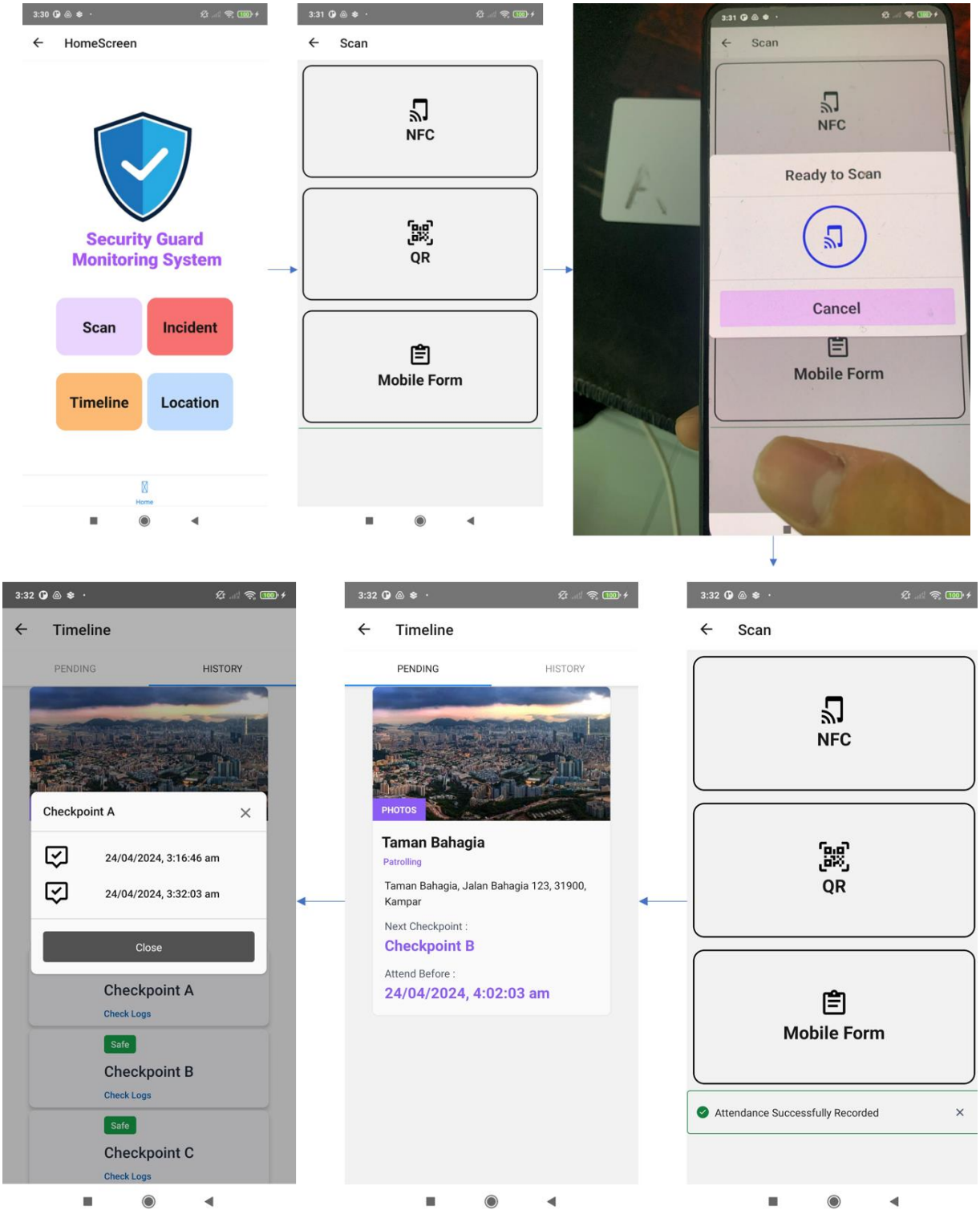


Figure 44 Test Case 11

Test Case 12: To test whether security guard can upload multimedia to report an incident successfully.

Preconditions: Login as a security guard (Jonny we created Test Case 1), navigate to security guard home page

Steps:

1. Click “Incident” button.
2. Click Camera icon.
3. Capture an image or upload an image in your device.
4. Click “Send Incident” button.

Expected outcomes: Notify admin and store the incident in the database.

Result:

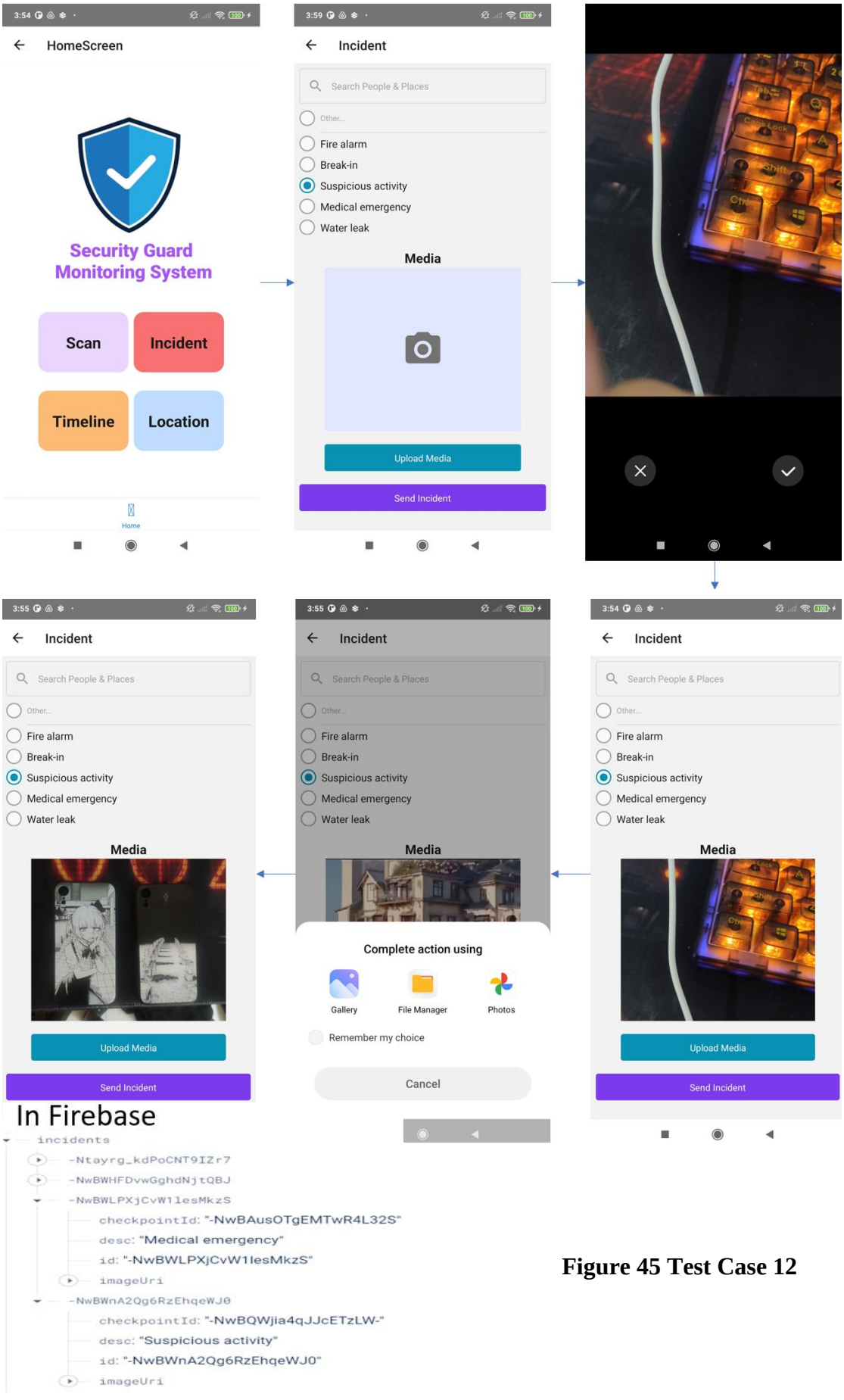


Figure 45 Test Case 12

References

Test Case 13: To test whether security guard can check his current location and assigned checkpoint on map successfully.

Preconditions: Login as a security guard (Jonny we created Test Case 1), navigate to security guard home page

Steps:

1. Click “Location” button.

Expected outcomes: Google Map with user current location and checkpoint location markers.

Result:

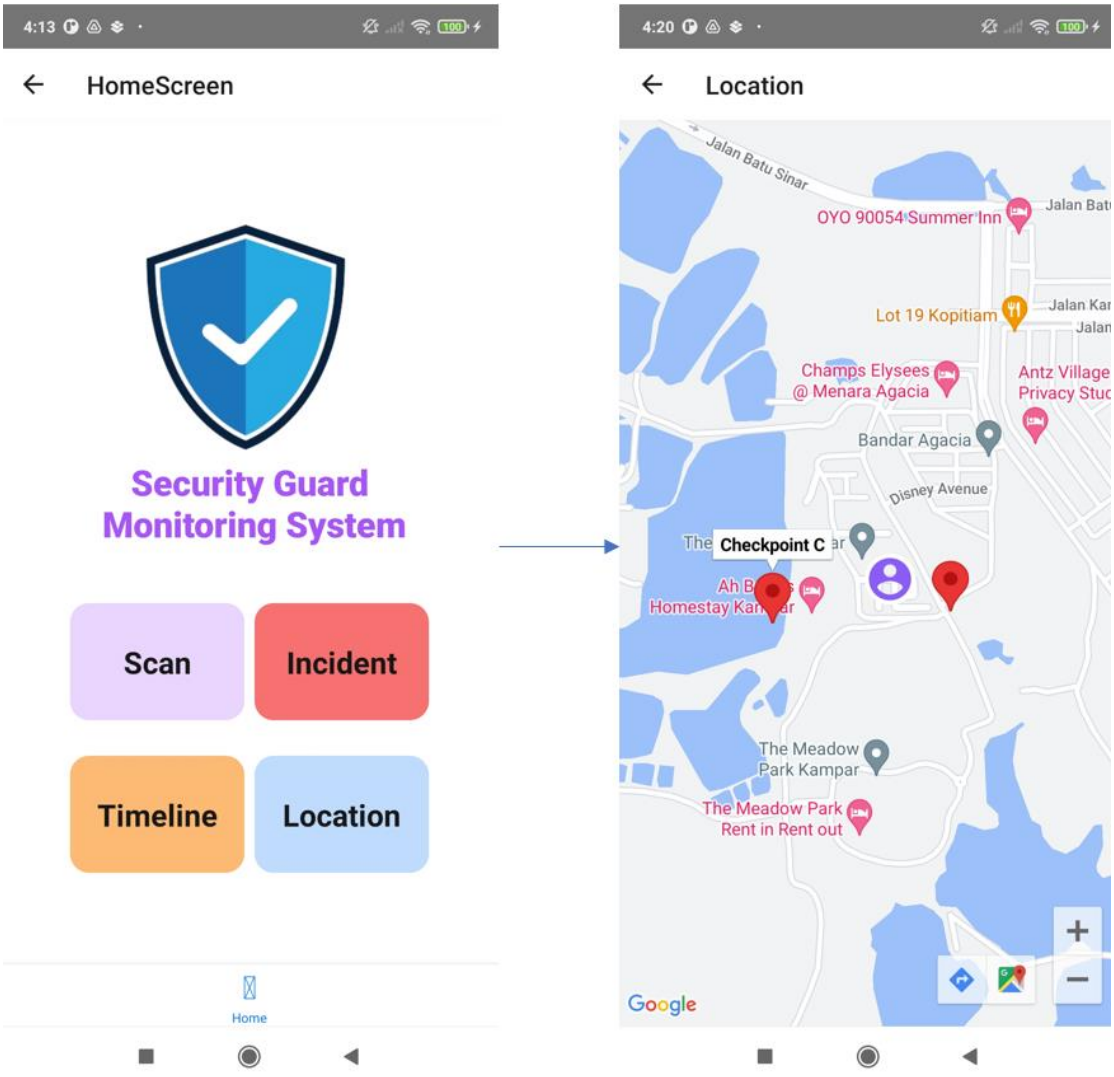


Figure 46 Test Case 13

References

6.2 Testing Setup and Result

Testing Environment:

Operating System: Android (Xiaomi Mi 9T as specified in 5.1)

IDE: Visual Studio Code

Device: Xiaomi Mi 9T as specified in 5.1 (Android)

Testing Data:

Predefined Users:

{

// Security Guard

Name: Jonny,

Email: jonny@gmail.com

Role: User

},

{

// Administrator

Name: LEE JIE LUN,

Email: jl@gmail.com,

Role: Admin,

}

Tools and Libraries:

Framework: React Native

UI Components: NativeBase

Database: Firebase Realtime Database

References

Testing Result:

Test Case 1:

Outcome: Pass

Observation: An account with email “jonny@gmail.com” and name “Jonny” is successfully created and login. It then navigates to security guard home screen.

Test Case 2:

Outcome: Pass

Observation: An account with email “jl@gmail.com” as “admin” role successfully login and navigates to Admin home page.

Test Case 3:

Outcome: Pass

Observation: An account with email “jonny@gmail.com” as “user” role successfully login and navigates to Security guard home screen.

Test Case 4:

Outcome: Pass

Observation: Role of Jonny is changed to “admin” and then successfully login and navigates to Admin home page.

Test Case 5:

Outcome: Pass

Observation: An new area has been added to the area list.

Test Case 6:

Outcome: Pass

Observation: A new checkpoint has been added under the area created in Test Case 5.

Test Case 7:

Outcome: Pass

Observation: The system successfully encrypted the data and write it into the NFC tags.

Test Case 8:

Outcome: Pass

Observation: The directed link shows what is the webcam capturing.

Test Case 9:

Outcome: Pass

Observation: The NFC tags UID can be scanned and added into the list of NFC UID.

Test Case 10:

Outcome: Pass

Observation: The admin successfully creates a assignment for security guard “Jonny” and display in Jonny’s security guard page.

Test Case 11:

Outcome: Pass

Observation: The system successfully decrypted the data stored in NFC tags and found out the correct checkpoint and area, then add logs under it.

Test Case 12:

Outcome: Failed

Observation: It shows that it successfully updates the database but did not notify the admin.

Test Case 13:

Outcome: Pass

Observation: A google map with security guard current location and checkpoint’s location markers is successfully retrieved and show to user.

6.3 Project Challenges

The project encountered challenges related to integrating multiple technologies and services, including NFC technology with encryption, surveillance camera integration, Google Maps API for displaying user and checkpoints' location, and managing the complexity of the database system. Implementing NFC technology with encryption required ensuring data security and integrity on NFC tags. The study on both encryption technology and NFC technology is needed to ensure the integration between these two feature is not conflict. Integrating surveillance camera feeds involved finding a most suitable application for hosting the webcam (as a replacement of surveillance camera) and the method to host the webcam video on public access. Utilizing the Google Maps API for location tracking demanded efficient handling of real-time location updates and optimal map rendering on mobile devices. Managing the complexity of the database system, likely Firebase Realtime Database, involved designing a well-organized relationship between entity to support real-time data synchronization and complex querying. Overcoming these integration complexities needs much effort to achieve these functionalities across all integrated components.

Managing dependencies on external services and frameworks, such as Firebase Cloud Messaging, give challenges in maintaining system stability and functionality throughout the project's development. Whenever install a new library through npm installation, there was a need to be aware to potential compatibility issues with the existing environment. To deal with the risk, a backup was copied before integrating any new library or dependency. This included maintaining version control to facilitate quick recovery in case of compatibility issues or unexpected failures come from library updates.

Other than those successful implementations, we also need to focus on the implementation that is failed. There are 2 main features failed to implement which are push notification with Firebase Cloud Messaging (FCM) and public access of webcam. There is a need to refine the push notification and integrate it with the Firebase Cloud Messaging. The reason it must be integrate with FCM but not using local notification is that the push notification is mainly for warning of scheduled time or new data of Incidents uploaded. These triggers can only be invoked if only the listener is always online and wait for events, but local notification only working when

the application is running, that is why the feature needs FCM to be integrated. On the other hand, the webcam which act as a surveillance camera in this project is only available for private network access. Meaning that only the devices under the same network can check on the camera which make the admin must be in the same geographical area with the security guards for monitoring current situation. This is not a good practice because an admin can be managing many areas and of course all the areas is not within the same geographical area. Such that, this feature must be improved and admin can access the webcam video everywhere in the future to achieve a better flow.

6.4 Objectives Evaluation

The implementation of NFC technology has significantly reduced the risk of tampering within the system. By utilizing NFC tags at checkpoints, the system retrieves accurate and reliable information regarding security personnel's movements, including date, time, location, and personnel details. And most importantly, the data inside the NFC tags is being encrypted and only verified NFC tags can be used to decrypt the data and update the Cloud Database. This feature has effectively enhanced the integrity of patrol records, resolving the challenge of tampering encountered in traditional monitoring systems.

The implementation of real-time log movement tracking has successfully minimized instances of missing patrols. By leveraging NFC technology and logging, the system can display the scheduled time for next checkpoint, and display the next checkpoint to the security guard. This decreases the chance of them forget to patrol around the time. However, the project faced challenges in implementing notification features using Firebase Cloud Messaging. This failure means that security guards are not reminded of scheduled patrols, and administrators are not alerted when patrols are missed for long period. This aspect requires improvement in future of the project.

The objective to integrate the monitoring system with existing infrastructure, including surveillance cameras, has been partially achieved. The system allows access to surveillance camera feeds associated with specific checkpoints, enhancing situational awareness and monitoring capabilities. However, there may still be room for improvement since the surveillance camera feeds is only working in a private network environment and failed to access by public network. Such that, the geographical distance between the administrator and the managing area is limited. This feature must be enhanced and enable public access with security key or method to ensure privacy.

6.5 Concluding Remark

In conclusion for chapter 6, the system testing phase has provided valuable information about the functionality of the Security Guard Monitoring System. Through a series of test cases, we have successfully validated key functionalities such as user registration, role-based authentication, NFC tag interactions, database operations, and integration with webcam (as replace of surveillance camera).

Most of the testing result is given a successful outcome, including the creation and authentication of user roes (“admin” and “user”), successful integration of NFC technology with encryption of data for checkpoint monitoring, and efficient management of areas, checkpoints, and assignments of the system. The NFC tags has increased the data integrity and security, decrease the risk of tampering of patrol records.

However, the evaluation process also reveals the area that needs further improvement. The features such as push notifications and public access for webcam has to be highlighted for future refinements. Addressing these challenges will be crucial to enhance system functionality, user engagement, and overall efficiency.

Moving forward into the key objectives’ evaluation. In this project, it successfully achieved objective 1 which is to avoid tampering by implementing NFC technology integrate with encryption technology. But only partially achieved for objective 2, which can only display scheduled time and next checkpoint for security guards and do not give a notification to them. Same to the objective 3 is also partially achieved, since the webcam can only be access in the private network, it could work for some situations, but it must not be the best practices. So, the efforts should be focusing on pushing objectives 2 and 3 to create a more completed system.

In summary, the evaluation and testing phase has given a good concept for future iterations and enhancements of the Security Guard Monitoring System.

CHAPTER 7 CONCLUSION AND RECOMMENDATION

7.1 Conclusion

In this project, the focus was on solving the security concerns associated with traditional and current security guard monitoring systems. The challenges met by these systems are tampering, missing patrol and absence of centralized control. These issues highlighted that they need an effective security guard monitoring solution. Besides, ensuring the safety of security guard personnel and assets in the areas became the primary motivation for the development of an improved security monitoring system.

Efficiency in security guard monitoring systems was targeted as a crucial motivation. The current existing challenges showed the importance of real-time tracking and reporting to adapt to the rapidly evolving security landscape. The requirement for a system ability to provide fast and accurate information became critical to improve overall security operations.

The proposed solutions given focused at addressing the challenges stated by implementing NFC technology to reduce tampering problems. The addition of log movement features aimed to reduce the number of missing patrols, ensuring the integrity of the system. In addition, the integration with existing infrastructure for centralized control was suggested to implement and improve overall system management.

The novel ideas introduced in this project included the integration of NFC technology, real-time reporting using Firebase Realtime Database to update the information and also using Firebase Cloud Messaging to notify users, and the utilization of the Google Map API for tracking security guard personnel's location. These proposed ideas contribute to the improvement of accuracy, efficiency, and response time for the security guard monitoring system.

In this report, it also shows the overall system design of the system, which included Entity Relationship Diagram(ERD), timeline and use case diagram, bring out a good foundation for the development of an effective security guard monitoring system. The ERD state the relationship between system entities. The use case diagram

and timeline give a clear roadmap of functionality and project progression respectively. These design can lead to a successful project implementation.

An effective system design contributes significantly to the smooth and efficient progress of the development phase. A comprehensive block diagram has been meticulously crafted for this system, accompanied by detailed specifications for each component, ensuring that no critical features are overlooked. Each component is thoroughly dissected, and individual flowcharts have been developed to illustrate the operation of every element. This meticulous approach guarantees clarity and comprehension of the system's flow and design.

A good system design can help the development phrase running smoothly and efficiently. A comprehensive block diagram is created for this system, accompanied by detailed specifications for each component, ensuring that no critical features are ignored. Every component is also further break down into a. This ensures that the system flow and design are clear and easy to understand.

The environment setup, featuring Visual Studio Code and Android Studio with essential extensions, which can bring me to efficient coding environments. The development toolkit's incorporation of React Native, NativeBase, and React Navigation enables a streamlined, cross-platform mobile application development process. The system users can separate into 2 different roles and providing different functionalities to interact with the system. For example, “admin” role doing most of the CRUD task for essential entities like User, Area, Assignments, and NFC tags UID. “user” role is mostly taking attendance and get information from database and the system.

Throughout its development, the project generated useful insights and lessons learnt. These observations include the research process, the issues encountered throughout system development, and the techniques used to overcome difficulties. The experience can help the developer overall development by offering practical knowledge in system design and implementation.

7.2 Recommendation

Based on the results of the evaluation and testing, a number of recommendations can be made to improve the Security Guard Monitoring System. It is advised to start by addressing the difficulties that happened during installation, such as fixing problems with the integration of Firebase Cloud Messaging for push notifications and enhancing camera access for remote monitoring. Prioritising improvements to the user interface and experience can involve enhancing accessibility features, streamlining navigation, and optimising screen layouts. Future development efforts might also concentrate on including extra security measures, like capabilities for real-time incident reporting or biometric verification. For biometric verification, we can use AWS Rekognition as a cloud solution, and it is easy to use. Such that, it does not require a machine learning developer to involved in it.

System functionality can be further improved by exploring cutting-edge technologies such as machine learning by applying predictive analytics to job planning. In addition, machine learning can be used for human detection and motion detection to maximize the efficiency of camera without keep it on all the time. It is important to prioritize usability testing and continuous user learning to gather information to continuously improve the system and adapt it to user needs and preferences. These recommendations are intended to guide further rounds of the development process and improve the overall effectiveness and efficiency of the security personal monitoring system.

REFERENCES

[1] QR-Patrol (2016) [Online]. Available: https://www.qrpatrol.com/system

[2] G. VR Gannapathy, V. Narayanamurthy, S. K. Subramaniam, A. F. B. T. Ibrahim, I. S. M. Isa, and S. Rajkumar, "A Mobile and Web-Based Security Guard Patrolling, Monitoring and Reporting System to Maintain Safe and Secure Environment at Premises," International Journal of Interactive Mobile Technologies, vol. 17, no. 11, pp. 4–14, Jun. 2023.

[3] Karakaya, M., Şengül, G., & Bostan, A. (2016). "A Wireless Control System Based on Smart Bluetooth and iBeacon Technology for Auditing the Patrols." International Journal of Scientific Research in Information Systems and Engineering (IJSRISE), 2, 8-13.

[4] V. Coskun, B. Ozdenizci, and K. Ok, “The Survey on Near Field Communication,” Sensors, vol. 15, no. 6, pp. 13348–13405, Jun. 2015, doi: 10.3390/s150613348.

[5] “Why do you need a security guard patrol system?,” THERMS, https://www.therms.io/blog/why-do-you-need-a-security-guard-patrol-system/

[6] V. Kaushik, K. Gupta, and D. Gupta, "React Native Application Development," International Journal of Advanced Studies of Scientific Research, vol. 4, no. 1, 2019.

[7] M. K. Caspers, "React and Redux," in Rich Internet Applications w/HTML and Javascript, vol. 11, 2017.

[8] M. L. Despa, "Comparative study on software development methodologies," Database Systems Journal, vol. 5, no. 3, 2014.

[9] W. Danielsson, "React Native application development," Master's thesis, Linköpings Universitet, Sweden, 2016.

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.: 2**
Student Name & ID: LEE JIE LUN	**21ACB01925**
Supervisor: Ts Tan Teik Boon	
Project Title: Security Guard Monitoring System	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

I have recreated the current frontend UI, and make new design to Admin role page and security guard page with react native, nativebase

2. WORK TO BE DONE

In the following week, I will try to do authentication using Firebase Authentication.

3. PROBLEMS ENCOUNTERED

Since there is only UI update in this 2 weeks, there is going smooth and no problem encountered.

4. SELF EVALUATION OF THE PROGRESS

I should push myself for faster and more efficient development schedule to develop the logic and backend of this system

Supervisor's signature

Student's signature

Trimester, Year: Year 3 Sem 3	**Study week no.: 4**
Student Name & ID: LEE JIE LUN	**21ACB01925**
Supervisor: Ts Tan Teik Boon	
Project Title: Security Guard Monitoring System	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

I have started to develop my backend database using Firebase Realtime Database

2. WORK TO BE DONE

In the following week, I will start developing NFC features

3. PROBLEMS ENCOUNTERED

Although I am using react native, but I found challenge when using react-native-firebase library, so I proceed to change to use firebase/app library which is more suitable for web app.

4. SELF EVALUATION OF THE PROGRESS

The connection between frontend and backend is quite hard for me, and used up many times to solve it.

Supervisor's signature

lee

Student's signature

Trimester, Year: Year 3 Sem 3	**Study week no.: 6**
Student Name & ID: LEE JIE LUN	**21ACB01925**
Supervisor: Ts Tan Teik Boon	
Project Title: Security Guard Monitoring System	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

I have started to play with NFC feature, like read, write into the NFC tags. Implementing it into my system for security guard to scan it and take attendance. For admin to write information into the NFC tags.

2. WORK TO BE DONE

After this, I will look into Google Map API to show user his current location.

3. PROBLEMS ENCOUNTERED

This is the first time I am playing around with hardware together with software, it is hard when first try to read and write the data of NFC tags.

4. SELF EVALUATION OF THE PROGRESS

It is all good for the progress and I need to be more understanding on NFC features for further implementation.

Supervisor's signature

lee

Student's signature

References

Trimester, Year: Year 3 Sem 3	**Study week no.: 8**
Student Name & ID: LEE JIE LUN	**21ACB01925**
Supervisor: Ts Tan Teik Boon	
Project Title: Security Guard Monitoring System	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

I have started to develop Google Map API, and display the current location of user and checkpoints marker to security guard for easier navigating them to their assignments.

2. WORK TO BE DONE

After this, I will look into encryption of data and store into the NFC tags.

3. PROBLEMS ENCOUNTERED

At first, I am struggling on getting the Google Map API key since it is costing. But I have found a replacement method and solve it.

4. SELF EVALUATION OF THE PROGRESS

There are still a few features needed to implement in this project, I have to speed up my progress.

Supervisor's signature

Student's signature

Trimester, Year: Year 3 Sem 3	**Study week no.: 10**
Student Name & ID: LEE JIE LUN	**21ACB01925**
Supervisor: Ts Tan Teik Boon	
Project Title: Security Guard Monitoring System	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

I have tried to implement encryption into the NFC tags and providing another security feature which is checking NFC tags UID to ensure security guard did not simply duplicate the data of NFC tags installed on the checkpoints and tampering is occurred.

2. WORK TO BE DONE

After this, I will look into integration of application with existing infrastructure.

3. PROBLEMS ENCOUNTERED

At first, I found a encryption library for react-native and it is keep failing and luckily I found another one which successfully implemented into this project.

4. SELF EVALUATION OF THE PROGRESS

Upon this development, this increase my understanding in encryption which is useful for my future development in security aspect.

Supervisor's signature

lee

Student's signature

Trimester, Year: Year 3 Sem 3	**Study week no.: 12**
Student Name & ID: LEE JIE LUN	**21ACB01925**
Supervisor: Ts Tan Teik Boon	
Project Title: Security Guard Monitoring System	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

I have started integrating the application with my webcam to simulate the situation that surveillance camera installed on the checkpoints. And I am using Yawcam to host a IP address that share my capturing of webcam.

2. WORK TO BE DONE

After this, I will start my report and final check of my application logic

3. PROBLEMS ENCOUNTERED

The problem is that the webcam capturing video is only working for private network which limited the geographical distance between the admin and security guards.

4. SELF EVALUATION OF THE PROGRESS

It is disappointing that I cannot make the webcam for public access which serve as a very useful feature for admin to access to surveillance camera without problem of geographical distance.

Supervisor's signature

lee

Student's signature

References

Trimester, Year: Year 3 Sem 3	**Study week no.: 14**
Student Name & ID: LEE JIE LUN	**21ACB01925**
Supervisor: Ts Tan Teik Boon	
Project Title: Security Guard Monitoring System	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

I have started and finalize my report for submission for competition and FYP submission.

2. WORK TO BE DONE

After this, I will finalize my report and create powerpoint for presentation

3. PROBLEMS ENCOUNTERED

There is no big problem encountered in report writing.

4. SELF EVALUATION OF THE PROGRESS

Report writing let me found out some of the problem existing in the current application and fix it properly.

Supervisor's signature

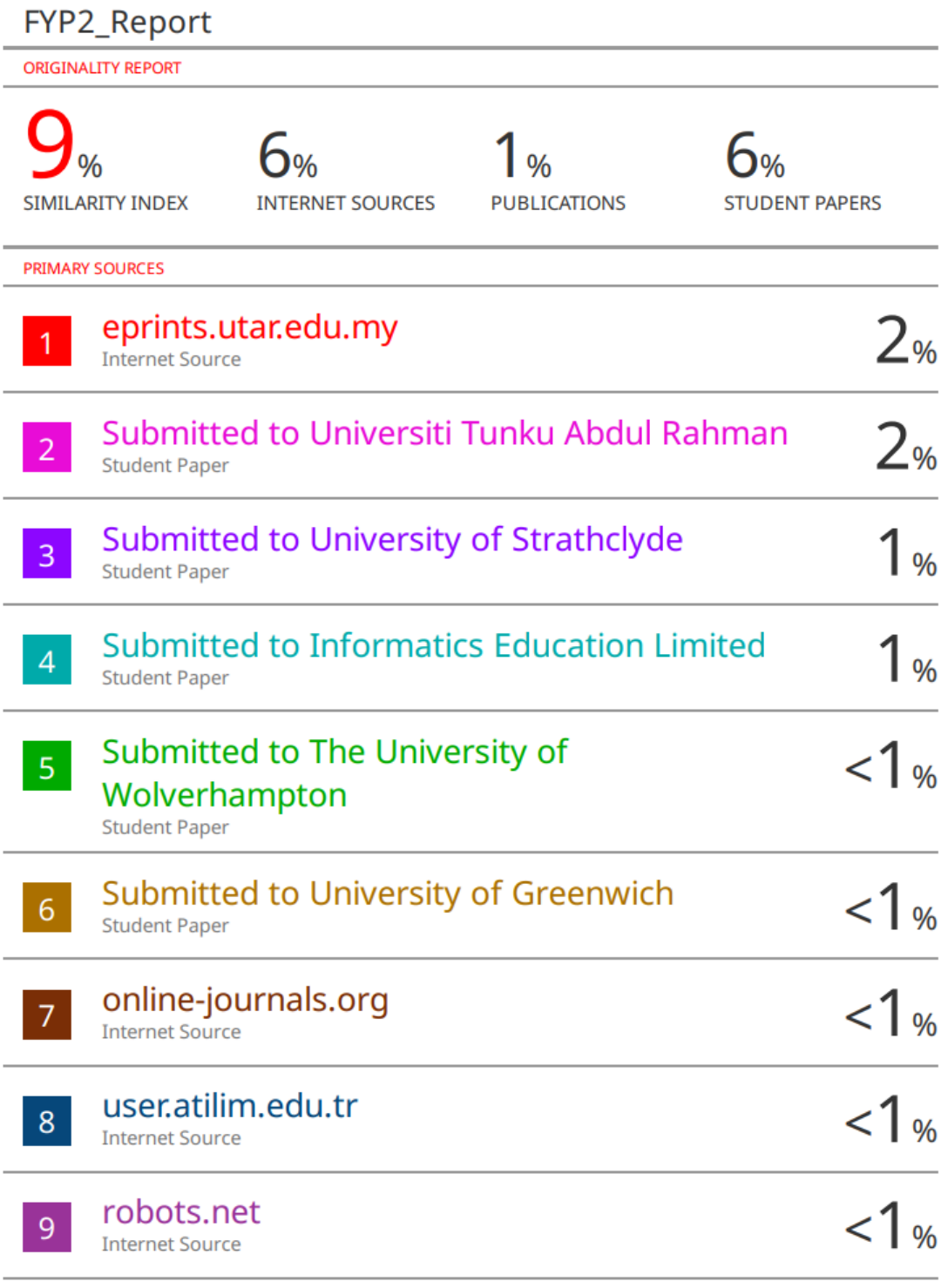
lee

Student's signature

POSTER


SECURITY GUARD MONITORING SYSTEM
WHAT IS THE PROBLEM OF EXISTING SYSTEM?
• Tampering Vulnerability
• Missing Patrols
• Absence of centralized control
OBJECTIVES
• Implement NFC Technology to Reduce Tampering
• Implement Log Movement to Minimize Missing Patrols
• Integrate with Existing Infrastructure for Centralized Control
CONTRIBUTION
• Enhanced Security Operations
• Integration with Surveillance Cameras
• Real-time Location Tracking
• Cross-Platform Development with React Native
Locating
ENVIRONMENT
• React Native
• NativeBase
• Firebase Realtime Database
• NFC technologies with encryption technology
• Google Map API Integration
• Multimedia Integration
• Private network access of surveillance camera
FUTURE WORK
• Push notification with Firebase Cloud Messaging on scheduled time and generate warning to administrators
• Public access of surveillance camera authenticate with security key

PLAGIARISM CHECK RESULT



FYP2_Report

ORIGINALITY REPORT

9%	6%	1%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	2%
2	Submitted to Universiti Tunku Abdul Rahman Student Paper	2%
3	Submitted to University of Strathclyde Student Paper	1%
4	Submitted to Informatics Education Limited Student Paper	1%
5	Submitted to The University of Wolverhampton Student Paper	<1%
6	Submitted to University of Greenwich Student Paper	<1%
7	online-journals.org Internet Source	<1%
8	user.atilim.edu.tr Internet Source	<1%
9	robots.net Internet Source	<1%

10	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
11	rdiot.tistory.com Internet Source	<1%
12	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
13	Submitted to School of Business and Management ITB Student Paper	<1%
14	Submitted to The University of Manchester Student Paper	<1%
15	Submitted to Federation University Student Paper	<1%
16	Submitted to University of Wolverhampton Student Paper	<1%
17	Submitted to UOW Malaysia KDU University College Sdn. Bhd Student Paper	<1%
18	repository.president.ac.id Internet Source	<1%
19	Submitted to University of Stirling Student Paper	<1%
20	github.com Internet Source	

		<1%
21	www.hindawi.com Internet Source	<1%
22	Submitted to The Hong Kong Polytechnic University Student Paper	<1%
23	eecs.ucf.edu Internet Source	<1%
24	library.polmed.ac.id Internet Source	<1%
25	www.atilim.edu.tr Internet Source	<1%
26	123dok.com Internet Source	<1%
27	digitalcollection.utem.edu.my Internet Source	<1%
28	mafiadoc.com Internet Source	<1%
29	thesai.org Internet Source	<1%
30	www.conceptdraw.com Internet Source	<1%

31	Submitted to Auckland University of Technology Student Paper	<1 %
32	Vedat Coskun, Busra Ozdenizci, Kerem Ok. "The Survey on Near Field Communication", Sensors, 2015 Publication	<1 %
33	Antonio Lazaro, Marti Boada, Ramon Villarino, David Girbau. "Battery-Less Smart Diaper Based on NFC Technology", IEEE Sensors Journal, 2019 Publication	<1 %

References

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	LEE JIE LUN
ID Number(s)	21ACB01925
Programme / Course	BACHELOR OF COMPUTER SCIENCE (HONOURS)
Title of Final Year Project	SECURITY GUARD MONITORING SYSTEM

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 9 % **Similarity by source** Internet Sources: 6 % Publications: 1 % Student Papers: 6 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: TAN TEIK BOON

Date: 25/4/2024

Signature of Co-Supervisor

Name: _____

Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	21ACB01925
Student Name	LEE JIE LUN
Supervisor Name	Ts Tan Teik Boon

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

Lee

_____

(Signature of Student)

Date: 22/4/2024