NEIGHBR:SMART RESIDENCE MANAGEMENT SYSTEM WITH RESIDENTIAL MANAGEMENT EFFICIENCY THROUGH AI-DRIVEN FACILITY FORECASTING AND DATA ANALYSIS

MUHAMMAD QAYYUM IMAN KHAIRIANTO

BACHELOR OF COMPUTER SCIENCE WITH HONOURS

2025

School Of Computing And Digital Technology

NEIGHBR:SMART RESIDENCE MANAGEMENT SYSTEM WITH RESIDENTIAL MANAGEMENT EFFICIENCY THROUGH AI-DRIVEN FACILITY FORECASTING AND DATA ANALYSIS

MUHAMMAD QAYYUM IMAN KHAIRIANTO

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at University Malaysia of Computer Science and Engineering (UNIMY) or other institutions.

Signature	:	
Name	:	**MUHAMMAD QAYYUM IMAN KHAIRIANTO**
ID No.	:	B09230018
Date	:	13/8/2025

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"NEIGHBR:SMART RESIDENCE MANAGEMENT SYSTEM WITH RESIDENTIAL MANAGEMENT EFFICIENCY THROUGH AI-DRIVEN FACILITY FORECASTING AND DATA ANALYSIS"** was prepared by **MUHAMMAD QAYYUM IMAN KHAIRIANTO** has met the required standard for submission in partial fulfilment of the requirements for the award of BACHELOR OF COMPUTER SCIENCE WITH HONOURS at University Malaysia of Computer Science and Engineering (UNIMY).

Approved by,

Signature :

Supervisor : PROF. HABIBOLLAH BIN HARON

Date : 13/8/2025

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to everyone who has played a role in the successful completion of the NEIGHBR project. This journey has been both challenging and rewarding, and it would not have been possible without the support and guidance of many individuals.

First and foremost, I extend my most sincere appreciation to my supervisor, Prof. Habibollah. His invaluable mentorship, constructive feedback, and unwavering patience were instrumental in guiding me through the complexities of this research. His expertise not only helped shape the technical direction of the project but also encouraged me to think critically and push the boundaries of my capabilities.

I am also incredibly grateful to my seniors, whose generosity in sharing their knowledge and past experiences provided me with essential guidance. Their willingness to point me toward the right references and offer practical advice saved me countless hours and helped me navigate technical hurdles with greater confidence.

Finally, this work is dedicated to my parents. Your endless love, sacrifices, and constant prayers have been the foundation of my strength. Knowing that I had your support gave me the motivation to persevere through every challenge. I am completing this degree for you, as a small token of my immense appreciation for everything you have done for me.

NEIGHBR:SMART RESIDENCE MANAGEMENT SYSTEM WITH RESIDENTIAL MANAGEMENT EFFICIENCY THROUGH AI-DRIVEN FACILITY FORECASTING AND DATA ANALYSIS

ABSTRACT

This project focuses on the development of NEIGHBR, a Smart Residence System, a mobile application designed to fortify residential security and optimize management operations through data-driven intelligence. Addressing the vulnerabilities of traditional manual access logs, the system introduces a dynamic, time-sensitive QR code framework for visitor management. This feature ensures high-level security by generating encrypted entry passes that automatically expire, effectively preventing unauthorized access and spoofing attempts. Beyond access control, the system integrates an AI-driven predictive analytics module designed to assist management in resource planning. By analyzing historical booking data, the system forecasts demand trends for shared amenities such as badminton courts and multipurpose halls, allowing administrators to proactively manage peak hours and prevent facility overcrowding. The application was developed using React-TypeScript and Supabase, ensuring a scalable architecture with strict adherence to the Malaysian Personal Data Protection Act (PDPA) 2010 through Row Level Security (RLS) policies. During the testing phase, the system underwent rigorous evaluation of its security protocols and forecasting accuracy. The results demonstrate that the system not only secures the premise through traceable digital logs but also empowers management with actionable insights, marking a significant advancement in secure, intelligent PropTech solutions for Malaysian communities

NEIGHBR: SISTEM PENGURUSAN KEDIAMAN PINTAR DENGAN KECEKAPAN PENGURUSAN KEDIAMAN MENERUSI RAMALAN FASILITI BERPACUKAN AI DAN ANALISIS DATA

ABSTRAK

Projek ini menumpukan kepada pembangunan NEIGHBR, sebuah Smart Residence System aplikasi mudah alih yang direka untuk memperkukuh keselamatan kediaman dan mengoptimumkan operasi pengurusan menerusi kecerdasan dipacu data. Bagi menangani kelemahan log akses manual tradisional, sistem ini memperkenalkan rangka kerja kod QR dinamik dan sensitif masa untuk pengurusan pelawat. Ciri ini menjamin keselamatan tahap tinggi dengan menjana pas masuk ternyahkripsi yang luput secara automatik, sekaligus menghalang akses tanpa izin dan percubaan penyamaran (spoofing) dengan berkesan. Selain kawalan akses, sistem ini mengintegrasikan modul analitik ramalan dipacu AI yang direka untuk membantu pihak pengurusan dalam perancangan sumber. Dengan menganalisis data tempahan lampau, sistem ini meramal trend permintaan bagi kemudahan bersama seperti gelanggang badminton dan dewan serbaguna, membolehkan pentadbir mengurus waktu puncak secara proaktif dan mencegah kesesakan fasiliti. Aplikasi ini dibangunkan menggunakan React-TypeScript dan Supabase, memastikan seni bina berskala dengan pematuhan ketat terhadap Akta Perlindungan Data Peribadi (PDPA) 2010 Malaysia melalui polisi Keselamatan Tahap Baris (Row Level Security - RLS). Semasa fasa pengujian, sistem ini telah menjalani penilaian rapi terhadap protokol keselamatannya dan ketepatan ramalan. Hasil dapatan menunjukkan bahawa sistem ini bukan sahaja menjamin keselamatan premis melalui log digital yang boleh dikesan, malah memperkasa pihak pengurusan dengan pandangan (insights) yang boleh diambil tindakan, menandakan kemajuan ketara dalam penyelesaian PropTech yang selamat dan pintar bagi komuniti di Malaysia.

TABLE OF CONTENTS

DECLARATION		iv
APPROVAL FOR SUBMISSION		vi
ACKNOWLEDGEMENTS		vi
ABSTRACT		vi
TABLE OF CONTENTS		x
LIST OF TABLES		xiv
LIST OF FIGURES		xv

CHAPTER	**TITLE**	**PAGE**
1	**INTRODUCTION**	**17**
	1.1 Background	17
	1.2 Problem Statement	18
	1.3 Research Question	19
	1.4 Research Objectives	19
	1.5 Scope of Research	20
	1.6 Thesis Outline	20
2	**LITERATURE REVIEW**	**23**
	2.1 Introduction	21
	2.2 Visitor Access and Security Challenges in Malaysian Residential Communities	24
	2.2.1 Related Research on Visitor Access and Digital Security Systems	24
	2.3 Application of Predictive Analytics in Residential Facility Management	25

	2.4	Artificial Intelligence Approaches in Residential Access Systems	26
		2.4.1 Application of Symbolic Artificial Intelligence in Smart Residential Management	26
		2.4.2 Artificial Intelligence and Predictive Analytics in Facility Management	27
		2.4.3 Comparison of Rule-Based Logic and Predictive Analytics in Facility Management	28
	2.5	Existing Platform for Smart Community Apps	29
		2.5.1 Local Studies on Smart Community Apps and Access Management	30
		2.5.2 Existing Applications Security Systems	30
	2.6	Summary	35
3	**METHODOLOGY**		**37**
	3.1	Introduction	37
	3.2	Research Design	37
		3.2.1 Agile Methodology Overview	37
		3.2.2 Agile Development Process for This Project	38
		3.2.2.1 Sprint 1 – Requirement Gathering & Analysis	39
		3.2.2.2 Sprint 2 – system Design & Prototype Development	39
		3.2.2.3 Sprint 3 –AI & Core Functionality Implementation	40
		3.2.2.4 Sprint 4 – Testing & Feedback Integration	

		3.2.2.5 Sprint 5 – Refinement & Final Documentation	40
	3.3	Experimental Setup	40
	3.4	Parameters	41
	3.5	Equipment	41
		3.5.1 User Interface Design Tool	42
		3.5.2 Backend Database & Authentication	43
		3.5.3 Integrated Development Environment (IDE)	44
		3.5.4 Mobile Application Framework	44
		3.5.5 Programming Language	45
		3.5.6 AI Development Language t	45
		3.5.7 Conclusion	46
	3.6	Summary	46
4	**DESIGN AND IMPLEMENTATION**		**47**
	4.1	Introduction	47
	4.2	Requirement	47
		4.2.1 Functional Requirements	48
		4.2.2 Non Functional Requirements	49
	4.3	Activity Diagram	50
		4.3.1 **Use Case 01**: Resident Authentication flow	50
		4.3.2 **Use Case 02**: Register Visitor & Generate QR	52
		4.3.3 **Use Case 03:** Facility Booking	50
		4.3.2 **Use Case 04**: AI Analytics & Facility Forecasting	54
	4.4	Entity Relationship Diagram (ERD)	58
	4.5	Graphical User Interface	59
		4.5.1 Admin Interface	59
		4.5.2 User Interface	64
	4.6	Coding Implementation	69
	4.7	Functionality Test Case	70
		4.7.1 Admin Functionality Test Case (Admin Part)	71
		4.7.2 User Functionality Test Case (Resident Part)	79

	4.8	Summary	85
5	**RESULTS AND DISCUSSION**		**86**
	5.1	Introduction	86
	5.2	User Acceptance Test (UAT)	86
	5.3	Demographic of User	87
	5.4	Conclusion of Findings	91
6	**CONCLUSION & RECOMMENDATIONS**		**92**
	6.1	Introduction	92
	6.2	User Acceptance Test (UAT)	92
REFERENCES			**94**
APPENDICES			**95**

LIST OF TABLES

TABLE	TITLE	PAGE
2.1	Comparison of Rule-Based Systems and Predictive Analytics (ML) in Resource Management	28
2.2	Comparison of Application Platform Summary	34
3.1	Gantt Chart of Smart Residence System	39
4.1	Intro Test Case	71
4.2	Register Test Case	71
4.3	Login Test Case	73
4.4	HompeageAdmin Test Case	74
4.5	FacilitiesAdmin Test Case	77
4.6	VisitorAdmin Test Case	78
4.7	ResidentLogin Test Case	79
4.8	ResidentHomepageTest Case	80
4.9	BookingFacility Test Case	82
4.10	VisitorResident Test Case	83
4.11	PofilePage Test Case	84

LIST OF FIGURES

FIGURE	TITLE	PAGE
2.1	INeighbour App Interface	32
2.2	JagaApp 2.0 Application Logo	32
2.3	MyTaman App Interface	33
3.1	Agile Methodology Map	37
3.2	Figma – User Interface Design Tool	42
3.3	Supabase - Backend Database & Authentication Service	43
3.4	Expo Go Logo	44
3.5	React Native Logo	44
3.6	TypeScript Logo	45
3.7	Python Logo	45
4.1	Use Case Diagram for Neghbr	48
4.2	Activity Diagram for User Registration and Login	50
4.3	Activity Diagram for Visitor Registration and Access	52
4.4	Activity Diagram for Facility Booking	53
4.5	Activity Diagram for AI Analytics Flow	55
4.6	Technical Data Flow for Prediction Logic	56
4.7	Entity Relationship Diagram for Smart Residence System	58
4.8	Smart Dashboard GUI	59
4.9	Booking Approval Interface	60
4.10	News Moderation Interface	61
4.11	Facility Insights & Analytics Interface	62
4.12	Security QR Scanner Interface	63
4.13	Smart Dashboard (User Interface)	64
4.14	User Profile	65
4.15	Facilities Page	66
4.16	News Page GUI	67

4.16	QR Generation page GUI	68
4.18	Snippet of code responsible for AI Prediction	69
4.19	Snippet of code Admin QR Scanner	69
4.20	Snippet of code for Database service function	70
5.1	General Usability Results visualised	87
5.2	App Layout Results Visualised	88
5.3	Navigation Ease Results Visualised	88
5.4	Readability of Text and Icons Results Visualised	89
5.5	Dark Mode Comfort Results Visualised	89
5.6	Visitor Management (QR code) results visualised	90
5.7	Login Process Results Visualisation	90
5.8	Facility Booking & Pricing Results Visualised	91

CHAPTER 1

INTRODUCTION

1.1 Background

In many residential communities, managing visitor access and shared facilities typically relies on manual verification by security personnel. While this traditional approach has served its purpose, it presents significant security vulnerabilities. Manual logbooks are easily falsified, physical checks are prone to human error, especially during peak hours, and static access protocols often fail to prevent unauthorized entry. As residential populations grow, the lack of real-time monitoring and immutable data trails makes these analog systems increasingly inadequate for ensuring the safety of residents.

The rapid advancement of mobile technology has introduced new standards for residential security and digital identity verification. Mobile applications now play a critical role in mitigating risks by replacing physical logs with encrypted digital records. According to the Malaysian Smart City Framework (MAMPU, 2022), adopting digital solutions is not just about convenience but is essential for creating 'Safe Cities' through data-driven surveillance and secure infrastructure.

In line with this vision, this project proposes the development of **Neighbr**, a security-centric mobile solution designed to fortify residential access control. Unlike traditional systems, **Neighbr** utilizes **dynamic, time-limited QR codes** and **verified management**

approval workflows to ensure that every entry and facility usage is authorized and traceable. This system aims to close the security gaps of manual processes, offering a robust, audit-ready platform that prioritizes the safety and privacy of the community.

1.2 Problem Statement

In many Malaysian residential communities, the management of visitor access and shared facility bookings remains predominantly manual, relying on physical security personnel, paper logbooks, and static intercom systems. These traditional approaches not only result in operational inefficiencies such as congestion at entry points but, more critically, expose the community to significant security vulnerabilities. According to Ramli et al. (2021), such systems are prone to verification loopholes and falsified records, especially during peak hours, which may lead to unauthorized access. Furthermore, Mohamad and Tan (2022) highlight that the overdependence on manual checks often results in untraceable access events, where security lapses go undetected due to the lack of immutable digital audit trails, thereby compromising the overall safety of residents.

Although digital solutions have been introduced in some communities, most existing systems lack intelligent security enforcement and predictive capabilities. Current applications are typically static in nature, functioning merely as digital forms without dynamic verification logic or data-driven decision support. Jamil et al. (2023) point out that many of these platforms still rely on reactive measures, rendering them ineffective in pre-empting security risks or managing resource demand. Additionally, Lee and Farhana (2022) report recurring issues with static QR codes being shared unauthorizedly and facility overcrowding, often due to the absence of dynamic encryption and forecasting algorithms that allow management to control usage proactively.

This lack of a secure, data-driven system poses a significant challenge for residential areas striving toward digital transformation, particularly in the context of Malaysia's smart city goals outlined in the MyDigital Blueprint (MAMPU, 2021). Without a predictive and security-first technological solution, communities risk operating in a reactive manner, where security breaches and facility overcrowding are addressed only after they occur, putting heightened pressure on security staff in densely populated zones.

Therefore, there is a critical need for the development of an intelligent mobile application that fortifies visitor management through dynamic QR code encryption and optimizes facility usage through AI-driven forecasting. Such a solution would not only streamline operations but also provide management with centralized control and predictive insights, ensuring access integrity and aligning with the nation's vision for safe, smart, and sustainable urban living.

1.3 Research Questions

To guide the development of **Neighbr**, a mobile application for managing residential access and facility bookings in Malaysia, this research aims to explore the following research questions:

1. How can a centralized digital access framework utilizing dynamic QR codes be implemented to enhance security oversight and visitor tracking for residential administrators?

2. In what ways can AI-driven predictive analytics assist residential management in forecasting facility usage trends to optimize resource allocation and reduce booking conflicts?

3. How can an AI-powered support system reduce management workload and improve resident communication?

1.4 Research Objectives

This project is guided by the following key objectives:

1. To investigate the use of **predictive analytics** and **QR code technology** as measures for optimizing facility resource allocation and digitizing visitor access control in a smart residential environment.

2. To develop a Smart Residence System Application that integrates **machine learning algorithms** for facility usage forecasting and **dynamic QR codes** for streamlined visitor management.

3. To evaluate the accuracy of the **facility demand predictions** and the effectiveness of the system in **enhancing operational efficiency**, streamlining tenant registration, and simplifying the visitor entry process.

1.5 Scope of Research

This project focuses on the development of **Neighbr**, a mobile application designed to **streamline access efficiency**, convenience, and facility management for residential communities, with the primary focus on properties located in the Petaling Jaya (PJ) area. The system serves both tenants and visitors, providing a secure and streamlined process for entry, booking, and communication with management.

The main scope of the application includes:

1. **Tenant QR Code Generation for Visitor Access** – Tenants can generate unique, time-sensitive QR codes for their visitors, enabling quick and secure entry without the need for manual registration with security personnel.

2. **Visitor Check-In Database** – All visitor entries are recorded in a centralised database accessible to management, ensuring accurate tracking of access history and enhancing overall security.

3. **AI-Driven Facility Analytics & Booking** – Tenants can seamlessly book shared facilities (gym, courts, BBQ), while the system utilizes predictive analysis to forecast future usage trends and estimate maintenance budgets for management.

1.6 Thesis Outline

Based on the objectives previously presented and the proposed approach, this thesis consists of five (5) chapters, the contents of which are summarised as follows:

1. **Chapter 1 – Introduction**:

   This chapter introduces the research background, emphasizing the critical security vulnerabilities and operational blind spots inherent in traditional manual management systems within Malaysian residential communities. It defines the problem statement, research objectives, and research questions, focusing on the need for access integrity and data-driven decision-making. The chapter introduces the developed solution, Neighbr, outlining its core capabilities in fortifying security through dynamic QR code encryption and optimizing resource allocation via AI-driven predictive analytics, ultimately aiming to transform residential management from a reactive to a proactive model.

2. **Chapter 2 – Literature Review**:

   The second chapter reviews existing literature relevant to smart residential security, digital access control protocols, and the application of machine learning in facility management. It examines the limitations of current market solutions, which often rely on static verification methods and lack intelligent data utilization. Comparative studies are presented to highlight the gap for a comprehensive, security-centric platform. The chapter concludes by validating the need for a system that integrates time-sensitive digital credentials with forecasting algorithms to address the specific density and security challenges of Malaysian residential zones.

3. **Chapter 3 – Methodology**:

   Chapter 3 details the research design and the Agile development methodology employed to build the system. It documents the full software development lifecycle (SDLC), describing the implementation of the React-TypeScript mobile architecture and the Supabase backend infrastructure. Special attention is given to the logic behind the dynamic visitor validation engine and the mathematical models used for the facility demand forecasting module. The chapter also outlines the experimental setup, including the specific testing parameters used to evaluate system latency, prediction accuracy, and security compliance (PDPA).

4. **Chapter 4 – Design and implementation**

This chapter details the comprehensive system architecture and the technical implementation of the Neighbr application. It defines the functional and non-functional requirements necessary to ensure the system's access integrity and forecasting accuracy. The chapter illustrates the system's operational logic through Use Case Diagrams (UCD) and Sequence Diagrams, specifically highlighting the secure data flow for dynamic QR code generation and the processing steps for the AI-driven facility prediction algorithms. Furthermore, it presents the actual coding implementation using React-TypeScript and Supabase, accompanied by the final Graphical User Interface (GUI) designs for both the resident mobile app and the management analytics dashboard.

5. **Chapter 5 – Results and Discussion**

This chapter presents the actual findings and performance data derived from the development and testing of the Neighbr system. It showcases the fully implemented features, providing a technical analysis of the visitor access workflow and the management dashboard’s predictive insights. The discussion focuses on quantitative results, such as the accuracy of the facility usage forecasts and the response times of the encrypted QR generation. Furthermore, it analyzes user feedback from testing sessions to evaluate the system’s effectiveness in mitigating unauthorized access and streamlining management operations compared to traditional manual methods.

6. **Chapter 6 – Conclusion and Future Work**:

The final chapter summarizes the project's key contributions, confirming the successful deployment of a secure, intelligent residential management system. It reflects on the system’s impact in closing security gaps and empowering management with actionable data. The chapter also discusses the limitations encountered during the development phase, such as external hardware integration constraints. Finally, it proposes recommendations for future enhancements, including the potential integration of IoT smart locks and advanced anomaly detection algorithms to further elevate the safety and efficiency of modern smart communities.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter reviews existing literature and practices relevant to residential access control, digital community management, and mobile technology integration in Malaysian housing developments. It draws from three key areas: academic studies on visitor access and security challenges in residential communities, local research on the adoption of smart building technologies and predictive data analytics in Malaysia, and comparisons of existing smart residential platforms used to manage access, bookings, and resident communication.

By combining these perspectives, this chapter identifies the current gaps in residential management systems, especially the lack of a unified, secure, and user-friendly mobile application tailored to Malaysian residents. These findings reinforce the need for a solution like the Smart Residence System which integrates predictive analytics and QR code technology for visitor registration and booking, into a single mobile platform for a more efficient and secure residential experience.

2.2 Visitor Access and security challenges in Malaysian Residential Communities

Visitor access remains a fundamental concern in Malaysian gated communities and condominiums. Most residential areas still depend on traditional methods such as paper logbooks, manual security checks, and intercoms. These outdated processes are vulnerable to human error, lack of verification, and false data entry, ultimately compromising safety and efficiency (Mohamad, Ramli & Aziz, 2021). Additionally, research shows that security personnel often lack proper training and professionalism, leading to resident dissatisfaction and communication breakdowns during access attempts (Azmi & Ismail, 2019).

Manual access systems also result in congestion during peak hours, inconsistent visitor screening, and limited traceability in case of incidents. As residential populations grow and urban areas become more densely populated, these inefficiencies have a direct impact on community well-being, especially when security personnel are unavailable or inattentive.

These issues underscore the importance of digitising access control using automated solutions, which not only streamline processes but also improve accuracy and safety in visitor management.

2.2.1 Related Research on Visitor Access and Digital Security Systems

Internationally, many studies have examined the growing role of smart technologies in managing building and residential access. Research shows that automation and contactless systems, specifically mobile-based credentials and dynamic QR codes, not only improve security but also significantly reduce administrative burden and enhance user convenience. Unlike biometric systems which raise privacy concerns, QR code-based solutions provide a secure, temporary, and verifiable method of entry that is easily adopted by users. Studies indicate that communities adopting digital visitor management systems report higher levels of resident satisfaction, largely due to streamlined entry processes and the elimination of physical registration queues.

In the Malaysian context, however, research on smart residential access systems remains limited. Most developments have focused on general smart city infrastructure, with little attention paid to visitor access protocols at the community level. A study by Mohamad, Ramli & Aziz (2021) noted that manual visitor check-in systems in gated communities are often unreliable, time-consuming, and easily bypassed. Moreover, the inconsistency in security guard performance and the lack of real-time verification have been repeatedly highlighted as common vulnerabilities in traditional residential management.

Despite the growing interest in property technology (PropTech) in Southeast Asia, there remains a lack of academic literature exploring the integration of end-to-end mobile visitor management within residential environments in Malaysia. This presents an opportunity for innovation, especially given the high smartphone penetration and increasing openness to digital solutions among Malaysian residents. The Smart Residence System aims to build upon global best practices by replacing manual logs with verifiable digital records, addressing the specific efficiency and security concerns faced by local communities.

2.3 Application of Predictive Analytics in Residential Facility Management

Predictive analytics has emerged as a transformative tool for resource optimization across various sectors, including hospitality, transportation, and property management. In the context of residential facilities, predictive models analyze historical usage data such as booking frequency, peak hours, and seasonal trends to forecast future demand. Unlike traditional static booking systems, AI-driven approaches utilize pattern recognition to anticipate facility overcrowding or underutilization, allowing management to optimize maintenance schedules and improve resource allocation strategies.

In Malaysia, the adoption of data analytics in real estate, often referred to as PropTech (Property Technology), is gaining traction. While early adoption was limited to commercial buildings, recent trends indicate a shift towards "smart condominiums" that leverage data to enhance the living experience. However, most local residential apps still rely on reactive management models, where issues are addressed only after they occur. By leveraging machine learning algorithms, residential management can transition to a proactive approach, predicting amenities' demand such as badminton courts or swimming pools before congestion occurs.

From a technical perspective, forecasting algorithms such as Time Series Analysis (e.g., ARIMA) or Regression Models can be integrated into mobile applications to provide real-time usage insights. This allows for the development of a cost-effective, intelligent management system that maximizes facility utilization without the need for expensive hardware sensors. The system can process booking logs to generate "busy level" predictions, helping residents plan their activities more effectively.

However, the implementation of such data-driven systems must strictly adhere to the Personal Data Protection Act (PDPA) 2010. While usage logs are less intrusive than biometric data, they still constitute personal behavioral data. Therefore, the Smart Residence System is designed with privacy-first principles, ensuring that data collection is transparent, anonymized where possible, and used strictly for the purpose of improving community services rather than individual surveillance (Yusof & Rahman, 2022).

2.4 Artificial Intelligence Approaches in Residential Access Systems

Artificial Intelligence (AI) plays a pivotal role in enhancing automation, operational efficiency, and resource optimization within modern smart residential systems. By leveraging different AI paradigms, ranging from symbolic, rule-based reasoning to data-driven machine learning, these systems can perform complex decision-making, automate booking approvals, and forecast facility usage patterns with high accuracy. The integration of AI not only streamlines resident and visitor management but also enables proactive maintenance and planning through predictive analytics. This section explores key AI approaches relevant to smart residential management, highlighting their principles, applications, and suitability for optimizing community operations.

2.4.1 Application of Symbolic Artificial Intelligence in Smart Residential Management

Symbolic Artificial Intelligence (Symbolic AI), also known as Good Old-Fashioned Artificial Intelligence (GOFAI), is a rule-based approach to AI that represents knowledge through explicitly defined symbols and formal logic. This methodology operates on the principle that intelligent behavior can be achieved by manipulating these symbols according to structured reasoning processes, allowing systems to make decisions and solve problems through predefined rules and heuristic strategies. Symbolic AI is particularly effective in domains where tasks can be clearly specified and logically decomposed, such as expert systems, regulatory compliance, and diagnostic decision-making. Its key advantage lies in its interpretability, as each decision can be traced back to a transparent set of logical rules an important factor in systems where accountability and auditability are essential.

However, the rigidity of Symbolic AI limits its ability to adapt to uncertain, incomplete, or highly dynamic real-world data. In contemporary AI development, Symbolic AI often serves as a component in hybrid architectures, combining its structured reasoning capabilities with the adaptability of statistical or machine learning models. Within the context of smart residential management, Symbolic AI can provide a transparent and accountable decision-making framework for processes such as visitor risk assessment, facility booking validation, and policy enforcement, ensuring both operational efficiency and regulatory compliance.

2.4.2 Artificial Intelligence and Predictive Analytics in Facility Management

AI-driven predictive analytics has emerged as the new standard for optimizing resource allocation and demand forecasting in smart management applications. Unlike static rule-based systems or traditional statistical methods, advanced AI models are capable of processing vast amounts of historical data, such as booking logs, timestamps, and seasonal utilization trends to forecast future facility usage with superior accuracy.

Modern predictive pipelines utilizing Artificial Intelligence go beyond simple trend analysis. These models are designed to recognize complex, non-linear patterns within the data, effectively mapping input variables (e.g., time of day, day of the week, public holidays) to predicted outcomes (e.g., expected occupancy levels). By leveraging intelligent forecasting architectures, the system can detect subtle temporal dependencies and seasonality that manual analysis might overlook, allowing for a deeper understanding of user behavior.

For real-world deployments, these AI models rely on structured training datasets to refine their predictive capabilities. The models process raw historical inputs to generate real-time "busy level" indicators, providing users with instant insights into facility availability. To ensure reliability, the AI's performance is continuously evaluated using validation metrics such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE), which benchmark the model's forecasts against actual usage data to drive ongoing improvements.

In the context of NEIGHBR, a Smart Residence System, the integration of an AI-powered predictive module enables the proactive management of shared amenities like badminton courts or gyms. By intelligently analyzing past booking behaviors, the AI model anticipates peak hours and suggests optimal booking times to residents, effectively smoothing out demand spikes. This data-driven approach not only improves resident satisfaction by reducing booking conflicts but also allows management to maximize the utilization of community assets through predictive intelligence rather than reactive measures.

2.4.3 Comparison of Rule-Based Logic and Predictive Analytics in Facility Management

Table 2.1: Comparison of Rule-Based Systems and Predictive Analytics (ML) in Resource Management

Feature	**Rule-Based Systems (Static)**	**Predictive Analytics (Machine Learning)**
Approach	Relies on explicit, pre-defined "If-Then" logic (e.g., "If slot is empty, book it").	Relies on statistical models that learn patterns from historical data (e.g., "Fridays at 5 PM are typically full").
Capability	Reactive; can only process current availability and basic constraints.	Proactive; forecasts future demand and anticipates congestion before it happens.
Adaptability	Rigid; cannot adapt to changing user behaviors without manual code updates.	High; continuously updates its logic as new booking data is collected over time.
Computational Load	Low; requires minimal processing power.	Moderate; requires training and inference steps, but feasible on modern cloud/mobile systems.

Best Use Case	Simple calendars and inventory checklists.	Optimization, load balancing, and smart suggestion systems for high-traffic facilities.

Table 1 compares two approaches to facility management, Traditional Rule-Based Systems and Predictive Analytics (Machine Learning). Rule-based systems rely on static, predefined constraints, offering simplicity and low computational cost, but they are limited to "reactive" management only responding to booking requests as they happen. In contrast, Predictive Analytics utilizes machine learning algorithms to analyze historical usage patterns, allowing the system to forecast demand and provide "proactive" suggestions (e.g., recommending off-peak hours). While predictive models require more processing power for training and inference, they offer significantly higher adaptability and efficiency, making them superior for optimizing shared resources in a smart residential environment.

2.5 Existing Platform for Smart Community Apps

The rise of smart community applications in Malaysia reflects a growing demand for digital solutions that streamline residential management, optimize access efficiency, and improve communication between residents and property managers. Existing platforms, both local and international, have successfully introduced foundational features such as QR-based visitor registration, e-billing, and static facility booking to reduce reliance on manual paperwork.

However, most current solutions remain limited in their scope of automation. They primarily function as digital record-keeping tools rather than intelligent management systems. Specifically, they lack advanced data analytics capabilities such as predictive demand forecasting for facilities and automated congestion alerts. Most platforms rely on "reactive" booking models where users simply book an empty slot without providing insights or recommendations based on usage trends. This section reviews existing platforms, focusing on their feature sets to identify the specific technological gaps primarily the lack of predictive intelligence that the proposed Smart Residence System aims to address

2.5.1 Local Studies on Smart Community Apps and Access Management

Although there are several mobile apps available for community management in Malaysia, few of them are supported by structured academic research or standardization. Popular apps like iNeighbour and JAGAAPP 2.0 have been adopted in selected residential areas to facilitate visitor registration, announcements, and billing. However, these platforms primarily function as static administrative tools. While they utilize QR codes for access, they generally lack intelligent decision-support features, such as predictive facility analytics or automated congestion management. Consequently, they remain reactive systems relying on manual inputs and distinct approval steps rather than proactive systems that optimize community flow.

A local study by Azmi & Ismail (2019) found that while digital access systems are an improvement over physical logbooks, they still face challenges related to operational efficiency when not integrated with broader management data. Furthermore, several residential communities continue to rely on informal channels like WhatsApp or phone calls to coordinate visitor access or book shared facilities. This fragmentation often leads to double bookings, inefficient facility tracking, and a lack of centralized data for management planning.

Privately developed apps such as MyTaman have explored lightweight solutions for neighborhood communication, but they often do not integrate core functionalities like real-time usage forecasting or seamless tenant-visitor linking. These early efforts demonstrate the market’s interest in digital transformation but also expose the limitations of current tools, which often digitize manual processes without adding computational intelligence.

As Malaysia continues to embrace smart technologies, there is a clear gap in providing a professional, reliable, and data-driven mobile app. The Smart Residence System is positioned to fill this gap by offering a feature-rich platform that goes beyond basic digitization, integrating predictive analytics to help residents and management manage visitor access and facility bookings more efficiently

2.5.2 Existing Applications

Several smart residential platforms have been developed to improve property management and resident experience. In Malaysia, commonly used platforms include iNeighbour, MyTaman, and JAGAAPP 2.0, each offering features like digital visitor registration, e-billing, and facility booking. These apps aim to reduce reliance on manual security practices while improving transparency between residents and property managers.

For example, iNeighbour provides foundational features such as QR-code-based visitor entry and e-documents for booking common facilities. However, its booking system is static, relying on simple calendar slots without predictive analytics to manage peak demand or suggest optimal usage times. Its user interface is also often reported as unintuitive, with a fragmented flow between visitor management and facility services (iNeighbour, 2022). Similarly, MyTaman, developed for Malaysian gated communities, offers useful push notifications and billing reminders but is limited in terms of intelligent automation; it functions primarily as a digital noticeboard rather than a proactive management tool (MyTaman, 2021).

Internationally, platforms like China's Hikvision and Cloudwalk incorporate advanced smart building features, including extensive IoT monitoring and AI-based surveillance. While technologically robust, these systems typically rely on expensive hardware infrastructure (sensors and cameras) and often raise concerns regarding data privacy and high implementation costs (Wang et al., 2020).

To better understand the strengths and limitations of current smart residential solutions, it is useful to examine widely used platforms in detail. The following review looks at both local and international applications, focusing on their key features, technological capabilities, and areas where they fall short specifically in predictive data utilization and user-centric design which the Smart Residence System aims to address.

INeighbour

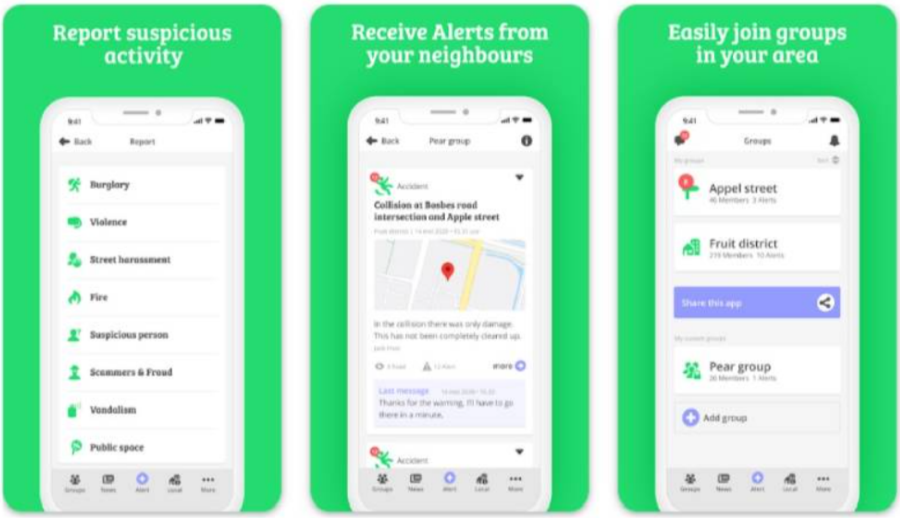


Figure 2.1 : INeighbour App

iNeighbour is a widely adopted smart community platform in Malaysia offering features such as QR-code-based visitor entry, e-billing, announcements, and facility booking. It integrates with intercom systems for added convenience and improves coordination between residents and security guards. However, it still requires manual approval for visitor entry and lacks a built-in facial recognition system, reducing its ability to provide seamless and fully automated access control.

<u>JAGAAPP 2.0</u>

Figure 2.2: JagaApp 2.0 Application Logo

JAGAAPP 2.0 is a mobile application designed for residential communities, focusing on visitor registration, guard communication, and basic billing functions. It allows residents to pre-register guests and generate QR codes for entry, successfully digitizing the traditional logbook process. Nonetheless, the platform operates primarily as a static administrative tool. It lacks intelligent decision-support features, such as predictive facility booking or automated congestion management. Consequently, facility usage remains a standard first-come-first-served process without data-driven insights to help management optimize resource allocation or resident scheduling

<u>MyTaman,</u>

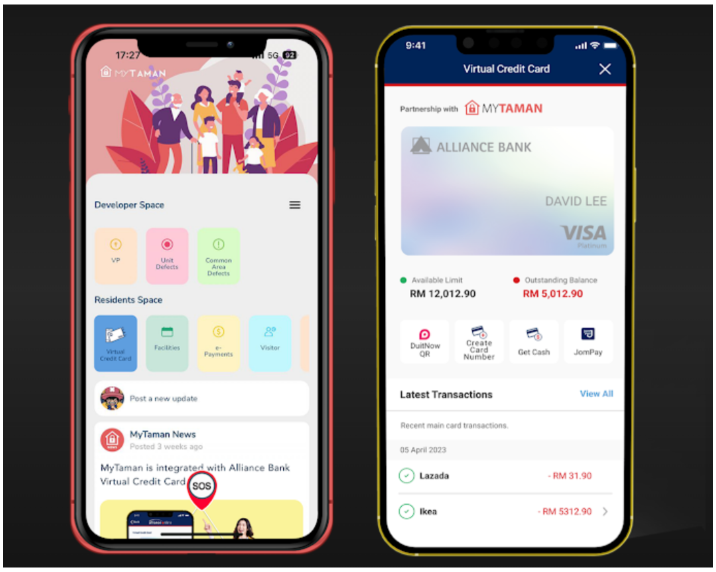


Figure 2.3: MyTaman App

MyTaman is a popular community management system in Malaysia that provides digital visitor check-ins using QR codes, announcements, billing statements, and facility bookings. While it successfully improves on traditional manual processes, the system operates primarily as a digital administrative tool. It lacks advanced data analytics capabilities, specifically predictive facility management. For instance, its booking system functions on a static first-come-first-served basis, offering no data-driven insights to help management forecast peak usage times or optimize amenity allocation.

On an international level, platforms such as ButterflyMX (USA) and CloudWalk (China) have implemented advanced features including AI-based access control and smart surveillance integration. However, these systems typically rely on capital-intensive hardware infrastructure (such as proprietary biometric scanners and smart intercoms) and complex surveillance grids. This high implementation cost, combined with potential concerns regarding data sovereignty and cross-border storage, makes them less feasible for widespread adoption in typical Malaysian residential settings compared to cost-effective, software-centric predictive solutions (Wang et al., 2020)

When compared to these platforms, the Smart Residence System sets itself apart by focusing on mobile-based facial recognition, real-time verification, and automated facility booking all in one application. It is designed to be accessible through smartphones without the need for QR scanning or physical forms, ensuring ease of use while maintaining a high level of security and data accuracy.

Table 2.2: Comparison of Application Platform Summary

Platform/System	**Core Features**	**Limitations**	**Relevance to Smart Residence System**
JagaApp 2.0	Visitor pre-registration via mobile app, QR code access, announcement board, security guard interface.	Operates as a **static system** without demand forecasting; lacks predictive analytics for facility usage; relies on manual booking slots without optimization suggestions.	Demonstrates successful digitization of visitor logs but highlights the gap for **intelligent facility management** and data-driven resource optimization proposed in this study.
i-Neighbour	Visitor management, facility booking, billing integration, intercom system, access card integration.	Booking system is **reactive** (first-come-first-served) with no AI-driven insights for peak hours; lacks proactive suggestions for residents; user interface often cited as complex.	Offers a multi-feature approach but remains a standard administrative tool. The proposed system improves upon this by integrating **predictive analytics** to optimize facility usage and streamline access.

MyTaman	Basic visitor management, guard notification, resident messaging, community announcements.	Lacks advanced facility booking logic and integrated data analytics; **no forecasting capabilities**; heavily dependent on manual guard verification for non-standard entries.	Validates the market demand for community apps but falls short of providing a comprehensive, **automated solution** that manages both access flow and facility efficiency intelligently.

Based on Table 2, several existing smart residence platforms like JagaApp 2.0, i-Neighbour, and MyTaman offer foundational features such as visitor management, facility booking, and resident communication. However, the comparison highlights notable limitations regarding computational intelligence and data utilization.

While JagaApp 2.0 and i-Neighbour successfully digitize access via QR codes, they operate primarily as static administrative systems. They lack predictive analytics to forecast facility demand or suggest optimal usage times, leaving facility management as a manual, first-come-first-served process. MyTaman, being the most basic among the three, serves as a lightweight communication tool but lacks integrated facility management and intelligent decision-support features altogether.

Overall, while these platforms provide useful digital solutions, they remain reactive, functioning mainly as digital logbooks. They fall short of offering a comprehensive, proactive system capable of using historical data to optimize community operations and streamline access flow, which is the core innovation of the proposed Smart Residence System.

2.6 Summary

Overall, the literature demonstrates that current visitor access and facility management systems in Malaysian residences are inefficient and often operate in silos. While there is growing interest in smart home solutions and data-driven decision-making, there remains a lack of unified, secure, and user-friendly platforms that combine these features specifically for the Malaysian context.

Most existing applications offer partial solutions digitizing forms without adding intelligence and are not optimized for full digital integration across all aspects of residential management. Privacy, ease of use, and efficient resource allocation remain major concerns. Therefore, this study proposes the development of a mobile-based Smart Residence System that addresses these limitations by offering a complete solution for QR-based access control, predictive facility management, and seamless communication tailored to the expectations and realities of Malaysian residential communities.

This chapter reviewed key literature and findings relevant to the development of a smart mobile application for residential access and facility management in Malaysia. Section 2.1 examined issues commonly faced in residential communities, including manual visitor registration, unreliable guard services, and outdated booking systems. It also explored the growing acceptance and feasibility of using predictive data models to optimize community operations.

Section 2.2 presented comparative analysis of current systems used in Malaysian residential settings. While some apps have introduced basic QR-based access, many still operate as static tools requiring significant manual intervention. The findings indicated a strong public interest in adopting a fully digital, AI-driven mobile app that not only simplifies processes but also uses data to forecast and optimize facility usage.

Section 2.3 and 2.4 highlighted the specific research gap in Malaysia's residential tech space. While international studies support the benefits of intelligent management systems, local research remains limited. Existing apps lack predictive capability and true automation, often functioning merely as digital logbooks. These gaps validate the need for a more comprehensive solution like the Smart Residence System, which combines predictive analytics with streamlined mobile access to provide a superior, user-friendly experience for both residents and management.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter outlines the methodology used for the design and development of the Smart Residence System. The project adopts the Agile software development methodology, which supports iterative development, continuous testing, and rapid adaptation to user feedback. Agile is particularly suitable for this project due to the evolving nature of its AI-based booking prediction and QR-based visitor management features, which require regular refinement to meet security, performance, and usability requirements.

3.2 Research Design

This chapter outlines the methodology used for the design and development of the Smart Residence System. The project adopts the Agile software development methodology, which supports iterative development, continuous testing, and rapid adaptation to user feedback. Agile is particularly suitable for this project due to the evolving nature of its AI-driven facility usage prediction and QR-based visitor management features, which require regular refinement to meet data accuracy, system performance, and usability requirements.

3.2.1 Agile Methodology Overview

Figure 3.1: Agile Methodology

The Agile methodology is a flexible, iterative approach to software development that emphasises collaboration, adaptability, and incremental delivery. Instead of following a rigid, linear plan, Agile promotes the continuous refinement of requirements and system features through short development cycles known as sprints.

This approach is well-suited to the Smart Residence System because the project involves user-centric features such as dynamic QR code generation for visitors and AI-driven facility usage forecasting, all of which benefit from frequent user feedback and ongoing improvements. Agile enables the development team to respond quickly to changes, incorporate new ideas from residents and management, and fine-tune predictive AI models as real-world booking data becomes available.

Key principles applied in this project include:

- Collaboration between developers, stakeholders, and future users to align the system with actual residential needs.

- Adaptability to accommodate evolving requirements, particularly for security measures and AI performance.

- Incremental delivery to ensure functional prototypes are available early for review and testing.

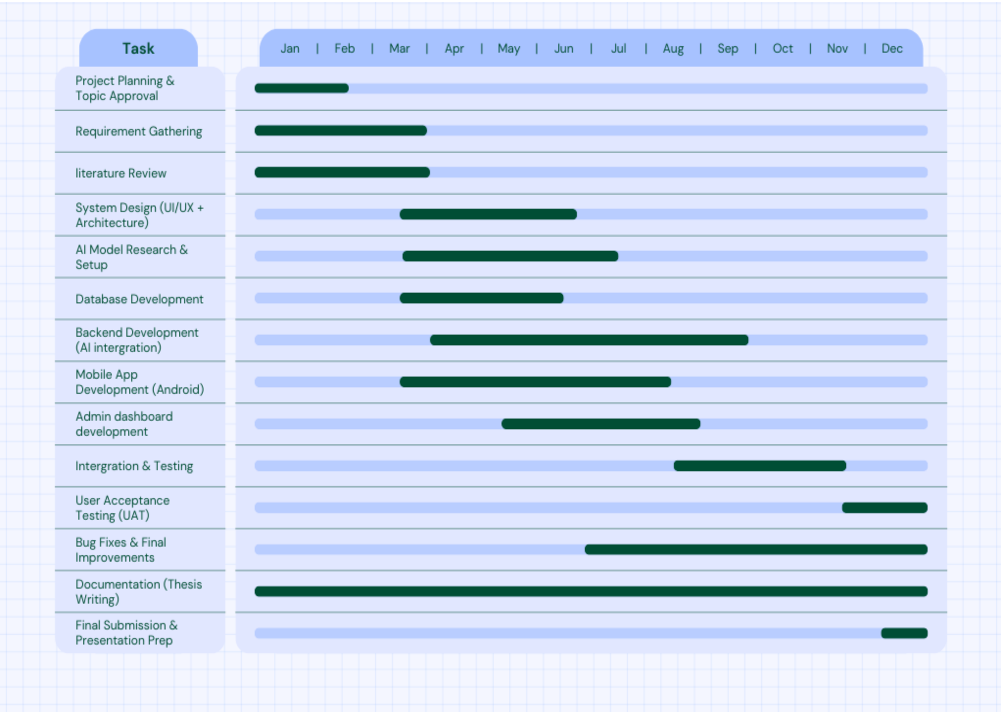
3.2.2 Agile Development Process for This Project

The proposed Smart Residence System will be developed using the Agile methodology, which emphasises iterative progress, flexibility, and continuous feedback from stakeholders. This approach is well-suited for mobile application development, where user requirements may evolve during the design and testing stages. By breaking the project into clearly defined phases, the development team can prioritise core functionalities, incorporate user feedback early, and refine features in short cycles to ensure the final product is both functional and user-friendly. The development process for this project is divided into five major phases as outlined below.

3.2.2.1 Sprint 1 – Requirement Gathering & Analysis

The first sprint focuses on identifying and documenting the key requirements for the Smart Residence System. Stakeholder meetings are conducted with building management and selected residents to understand current challenges in visitor management, booking systems, and billing integration. Requirements are grouped into major modules Facility Booking booking prediction and visitor management features. Functional requirements (e.g., QR code generation, AI-based booking prediction) and non-functional requirements (e.g., security, scalability) are also defined.

Table 3.1: Gantt Chart of Smart Residence System



3.2.2.2 Sprint 2 – System Design & Prototype Development

In this sprint, the system architecture and UI/UX design are developed. Tools like Figma are used to create interactive wireframes for the visitor interface and the admin dashboard. Data flow diagrams and process flowcharts such as the AI-based booking prediction and QR-based visitor management features validation flow, are produced to visualise system logic. A basic clickable prototype is developed for demonstration and early feedback.

3.2.2.3 Sprint 3 – Core Development & Algorithm Implementation

This sprint focused on the full-scale development of the application using React with TypeScript to ensure type safety and code scalability. The predictive analytics module was implemented using custom TypeScript algorithms that process historical booking arrays to calculate demand probabilities. Simultaneously, the dynamic QR code generation engine was integrated using React-based libraries to handle visitor link creation and validation. The management dashboard's booking filtering logic was also coded, enabling administrators to approve or reject requests based on real-time availability data.

3.2.2.4 Sprint 4 – System Integration & Testing

In this sprint, the developed React application underwent comprehensive integration testing. Test cases focused on the accuracy of the TypeScript-based forecasting logic (verifying if peak hours were correctly flagged based on the mock datasets) and the responsiveness of the QR code rendering. Cross-platform testing was conducted to ensure that the interface and visitor links functioned correctly on Android. Feedback regarding the responsiveness of the booking interface and the clarity of the 'busy level' indicators was collected and logged for optimization.

3.2.2.5 Sprint 5 – Refinement & Final Deployment

The final sprint involved refining the codebase, including code refactoring to optimize component re-rendering and improve app performance. The forecasting algorithms were fine-tuned to handle edge cases in the booking data. Final documentation was prepared, including API documentation for the backend connections and a user manual. The system was then finalized for demonstration, marking the completion of the development lifecycle.

3.3 Experimental Setup

The experimental setup involved the actual development environment of the Smart Residence System. The mobile application was built using the React framework with TypeScript, utilized for its component-based architecture and strong typing system which reduces runtime errors.

For the Predictive Component: The forecasting logic was coded in TypeScript and tested against a dataset of simulated historical bookings to validate the algorithm's logic. For the Visitor System: The QR code generation flow was tested on real mobile devices to measure generation speed and scanning validity. Tools Used: Visual Studio Code was used as the primary IDE, with Git for version control. Postman was used to test API endpoints for the management dashboard data flow.

3.4 Parameters

The following key parameters will be considered in the design:

1. **Algorithmic Accuracy:** Measuring the correctness of the TypeScript logic in flagging 'peak hours' compared to manual calculations of the dataset.
2. **QR Generation Latency:** The time taken (in milliseconds) for the React component to render a valid unique QR code upon form submission.
3. **Conflict Detection Rate:** The system's success rate in preventing overlapping facility booking requests during simultaneous user actions

3.5 Equipment

To develop the Smart Residence System, a mobile-based access and management application, a set of reliable and efficient tools has been selected to support the implementation of its core features, including facial recognition for secure entry, AI-assisted facility booking, integrated billing, and customer support servicesTo develop the Smart Residence System, a mobile-based access and management application, a set of reliable and efficient tools was utilized to support the implementation of its core features. These included dynamic QR code generation for visitor entry, predictive analytics algorithms for facility booking, and a management filtering dashboard. The selection of these tools ensured that the system was built on a scalable architecture capable of handling real-time data processing and cross-platform compatibility.

3.5.1 Figma – User Interface Design Tool

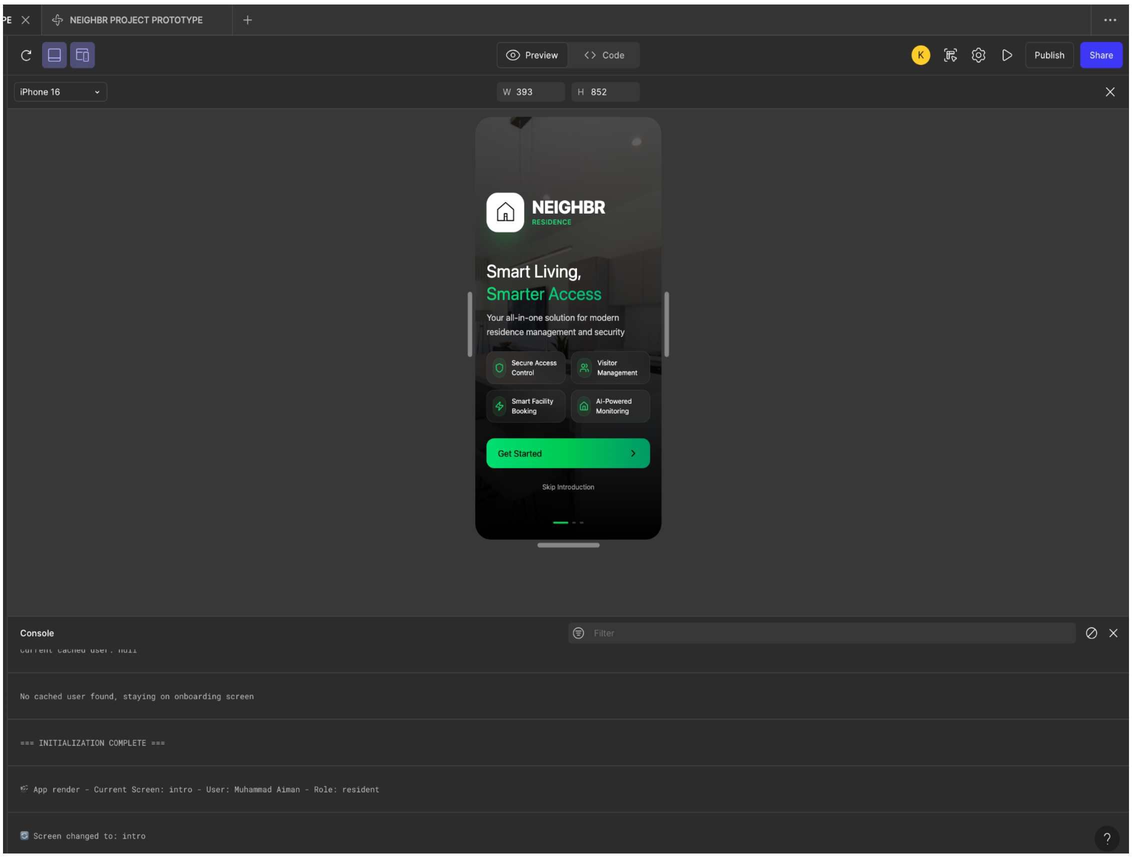


Figure 3.2: Figma

Figma served as the primary tool for designing and prototyping the user interface of both the mobile application and the admin dashboard. By creating interactive wireframes, the development team mapped out the navigation flow, visual hierarchy, and functional placement of key features such as QR code generation, predictive booking graphs, and the management approval interface. Using Figma at the design stage enabled stakeholders to visualise the application's structure early in the process, allowing for feedback and refinement before the React-TypeScript implementation phase began, which significantly reduced the need for code refactoring later

3.5.2 Supabase – Backend Database & Authentication Service

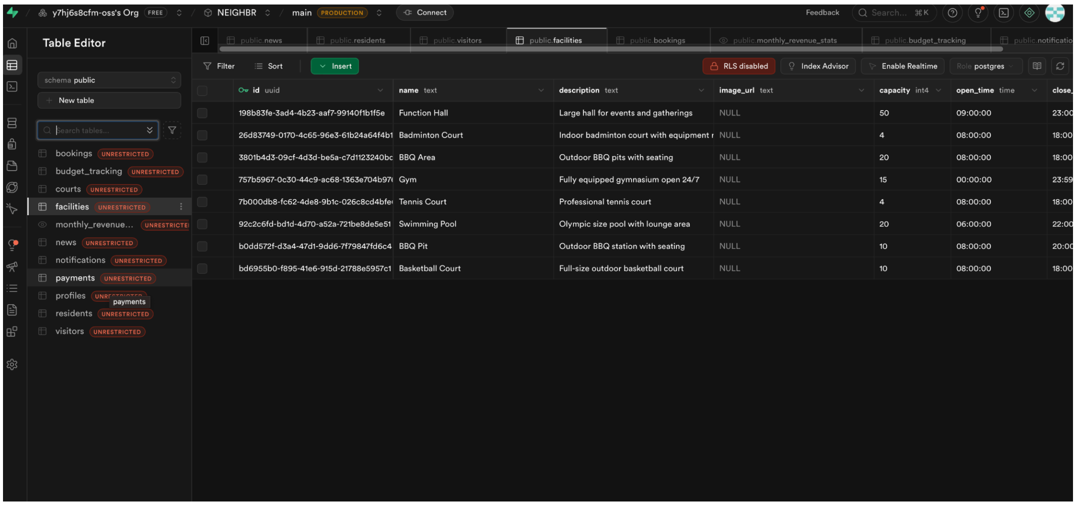


Figure 3.3: Supabase

Supabase was utilized as the primary backend infrastructure for managing the application's relational data and authentication. Built on top of PostgreSQL, its real-time capabilities made it highly suitable for securely storing resident profiles, visitor QR code logs, and facility booking schedules. During the development phase, the database was populated with structured datasets to simulate real-world usage, allowing for the rigorous testing of user authentication, booking conflict detection, and the retrieval of historical data required for the predictive analytics algorithms. Furthermore, Supabase's Row Level Security (RLS) policies were implemented to enforce strict access control, ensuring that residents could only access their own visitor records while management retained administrative oversight.

3.5.3 Expo Go – Integrated Development Environment (IDE)

Figure 3.4: Expo Go Logo

Expo Go will be employed as the primary development environment for building the mobile application throughout the development phase. By using Expo Go, the development team can efficiently write and debug code, instantly preview the user interface on physical devices or simulators, and utilize the Expo client for quick iteration and testing across various configurations without needing to build a native app for every change.

3.5.4 React Native – Mobile Application Framework

Figure 3.5: React Native Logo

React Native will be employed to develop the cross-platform mobile application throughout the development phase. By using React Native, the development team can build the application for both Android and iOS systems using a single codebase, ensuring a consistent user experience and utilizing features like "Hot Reloading" to accelerate the testing and refinement process before full-scale implementation.

3.5.5 TypeScript – Programming Language

Figure 3.6: TypeScript Logo

TypeScript will be employed as the primary programming language for writing the application code throughout the development phase. By using TypeScript, the development team can utilize static typing to detect errors during compilation rather than runtime, enhance code maintainability, and ensure safer data handling across the application before full-scale implementation.

3.5.6 Python – AI Development Language

Figure 3.7: Python Logo

Python will be employed to develop the artificial intelligence and machine learning components throughout the development phase. By using Python, the development team can leverage powerful libraries to train predictive models, process large datasets efficiently, and ensure the accuracy of the facility usage algorithms before full-scale implementation.

3.5.7 Conclusion

By combining **Expo Go** for interface design, **Supabase** for secure and scalable data management, **React Native** for Mobile Application Framework, and **TypeScript** for robust application development, the NEIGHBR was successfully transformed from a concept into a fully functional prototype.

The methodology adopted proved effective in addressing the project's core requirements. The **predictive analytics module** was successfully implemented and validated against historical datasets, while the **QR-based visitor access system** was tested for real-time responsiveness. Unlike the initial planning stage, the security and data privacy measures, specifically adhering to the **PDPA 2010** were actively integrated into the system architecture via Supabase's Row Level Security (RLS), ensuring a secure environment for all users.

3.6 Summary

This chapter detailed the design and development methodology of NEIGHBR, the Smart Residence System, outlining the specific tools, agile processes, and technical parameters used to build the application. It covered the implementation of key features, including the AI-driven facility forecasting, visitor QR access, and the management filtering logic. The proposed structure successfully solved the identified problems in residential communities through intelligent automation.

CHAPTER 4

DESIGN AND IMPLEMENTATION

4.1 Introduction

The first stage focuses on setting up the core functions of the Smart Residence System to establish a solid foundation for testing its performance. The main features, including facial recognition for tenant and visitor verification, QR code generation for visitor check-in, facility booking, and integrated billing display, will be tested in a simple environment to ensure each function operates as expected.

Once initial tests show stable performance and accurate results such as successful identity verification, correct booking confirmations, and smooth payment gateway redirection the system will be tested in more realistic residential scenarios. These will include situations like multiple concurrent facility bookings, high visitor traffic, and real-time communication between residents and management. This approach ensures the Smart Residence System is fully optimised to handle the day-to-day demands of a residential community while maintaining security, convenience, and efficiency.

4.2 Requirement

In this subtopic, the requirements of NEIGHBR mobile application will be divided into two sections and will be discussed in detail which are the functional requirements and non-functional requirements

4.2.1 Functional Requirements

In this subtopic, the functional requirements of the project will be discussed and a use case diagram will be constructed to explain the functionality of the residents (customers) and the administrator in the Neighbr mobile application in detail.

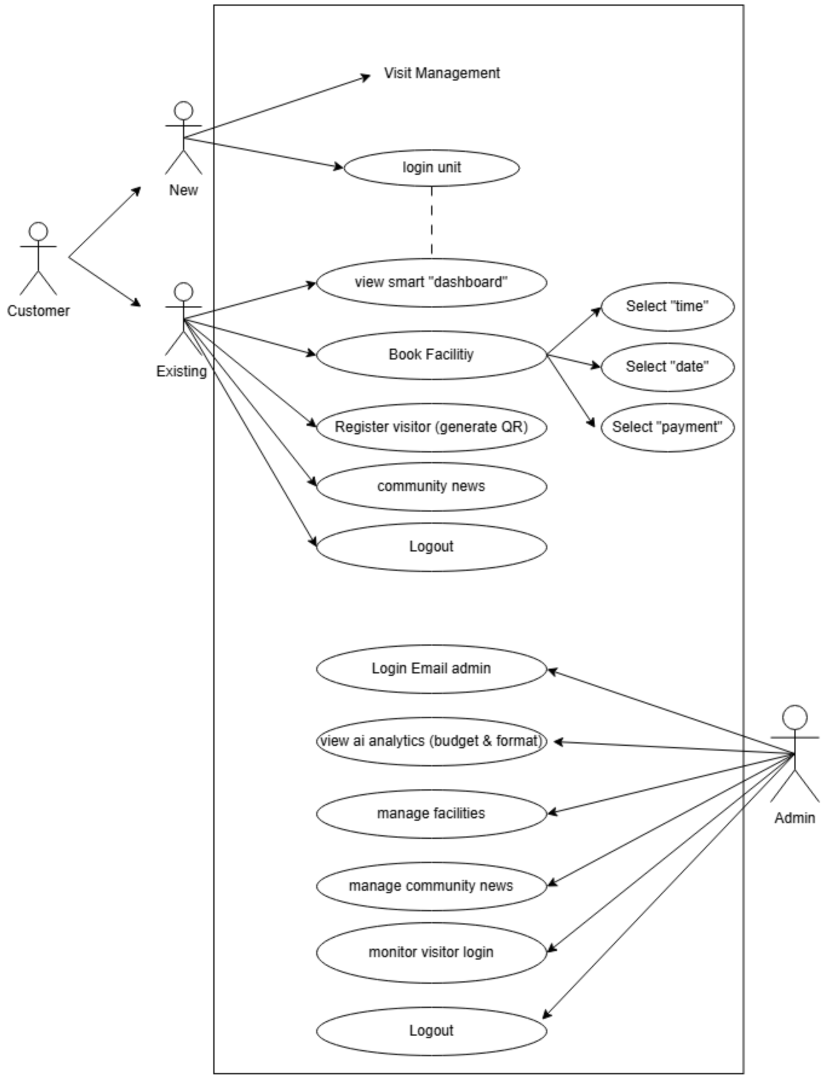


Figure 4.1: Use Case Diagram for Neighbr

Based on the use case diagram shown in Figure 4.1 above, the functional requirements of Neighbr are as listed below:

1. The system should enable the new and existing customers to log in to their user accounts.
2. The system should enable the users to view the smart dashboard upon logging in.

3. The system should enable the users to book facilities by selecting the specific date, time, and payment method.
4. The system should enable the users to register visitors and generate a secure QR code for access.
5. The system should enable the users to view community news and updates.
6. The system should enable the users to log out from the application.
7. The system should enable the admin to log in to the administrative account via email authentication.
8. The system should enable the admin to view AI analytics regarding facility usage and budget formatting.
9. The system should enable the admin to manage facility details and availability.
10. The system should enable the admin to manage and post community news.
11. The system should enable the admin to monitor visitor login logs for security purposes.
12. The system should enable the admin to log out from the admin account.

4.2.2 Non Functional Requirements

In this subtopic, the non-functional requirements, which define the quality attributes and performance standards of the system, will be discussed.

1. **Security.** The system should ensure that all sensitive personal data collected from residents and visitors such as names, contact numbers, and vehicle details are stored securely in compliance with the **Personal Data Protection Act (PDPA) 2010**. Additionally, the system must employ **encryption for the dynamic QR codes** to prevent unauthorized duplication or spoofing.
2. **Performance.** The system should ensure low latency in critical operations. specifically, the **QR code generation** for visitor access should occur instantly (within milliseconds) upon form submission to prevent delays at the guardhouse. Similarly, the **AI-driven facility forecasts** should load real-time availability data without significant lag.
3. **Reliability.** The system should ensure consistent availability, particularly for the **visitor access module**, as any downtime could result in physical entry barriers for

guests. The database (Supabase) must reliably handle concurrent booking requests without data conflicts or crashes.

4. **Scalability.** The system should be capable of handling an increasing number of resident profiles and historical booking datasets without performance degradation, allowing the **predictive AI model** to remain accurate as the community's data grows over time.

4.3 Activity Diagram

The activity diagrams for the Neighbr Smart Residence System illustrate the sequential flow of activities between the stakeholders (Residents and Administrators) and the system for the project's core functionalities. These diagrams visually map the step-by-step logic for each major operation, clarifying how the system handles security verifications and data processing. The following main use cases are defined and detailed in this section: User Login, Register Visitor (QR Generation), Book Facility, Admin Login, View AI Analytics, and Monitor Visitor Logs.

4.3.1 Use Case 01: Resident Authentication Flow (Login Module)

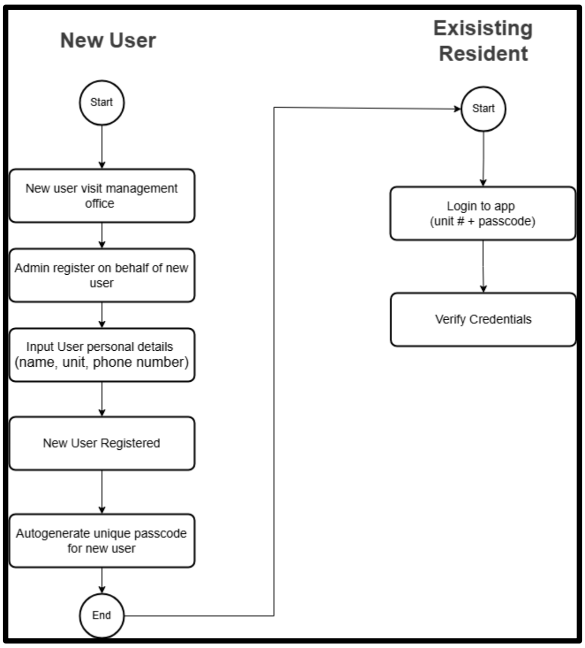


Figure 4.2: Activity Diagram for User Registration and Login

In the User Registration and Login use case, as shown in **Figure 4.2**, the process is divided into two distinct phases: the **Admin-assisted Registration** for new users and the **App Login** for existing residents. Unlike standard self-registration apps, **Neighbr** prioritizes security by requiring new users to visit the management office for identity verification before an account is created.

The system performs the following actions and validations during this workflow:

1. **Admin Verification & Data Entry:** The process begins when a new user visits the management office. The administrator validates the resident's physical documents and logs into the admin portal to register the user on their behalf. The admin inputs the required personal details, specifically the **Resident Name**, **Unit Number**, and **Phone Number**, ensuring that the unit number corresponds to a valid property within the database.
2. **Unique Passcode Generation:** Upon successful data entry, the system triggers an automated security function to **autogenerate a unique passcode** for the new user. This ensures that passwords are not weak or user-created but are system-assigned for higher security. This passcode is securely stored in the **Supabase** database linked to the specific Unit Number and provided to the resident offline.
3. **Credential Verification (Login):** For the login phase, the existing resident launches the application and enters their **Unit Number** and the system-generated **Passcode**. The system queries the Supabase *residents* table to verify if the combination matches an active record.
4. **Session Authorization:** If the credentials match, the system verifies the user's status and grants access to the dashboard. If the Unit Number does not exist or the Passcode is incorrect, an error message is displayed, prompting the user to re-enter their details or contact management.
5. **Data Integrity Check:** Throughout the registration process, the system checks for duplicate entries (e.g., ensuring a Unit Number isn't already assigned to a different primary owner without authorization) to maintain data integrity within the community records.

4.3.2 Use Case 02: Register Visitor & Generate QR Code

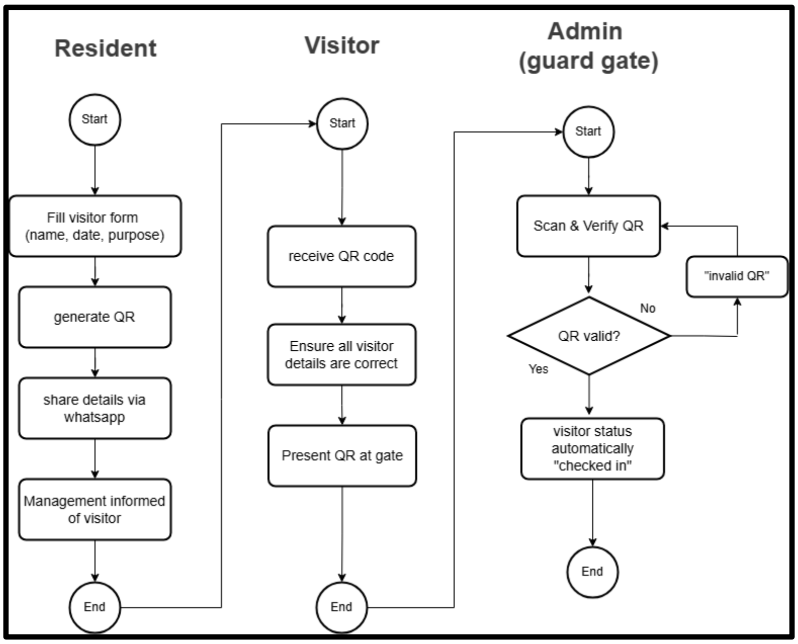


Figure 4.3: Activity Diagram for Visitor Registration and Acces

The Visitor Registration and Access use case, as illustrated in Figure 4.3, details the secure workflow for granting temporary access to non-residents. This process involves coordination between three key actors: the Resident, the Visitor, and the Admin (Guard Gate), ensuring that every entry is authorized and digitally tracked.

The system performs the following actions and validations during this workflow:

1. **Visitor Details Entry:** The process begins with the Resident initiating the request within the app. The resident fills in the visitor form with critical security details, including the visitor's Name, Intended Date of Visit, and Purpose. This data is logged to ensure an audit trail exists for every guest.
2. **Secure QR Generation & Sharing:** Upon form submission, the system validates the input and generates a unique, encrypted QR code. Simultaneously, the system updates the management dashboard ("Management informed of visitor") to alert security personnel of the incoming guest. The resident then shares this QR code directly with the Visitor via external messaging apps like WhatsApp.
3. **Visitor Verification (At the Gate):** When the Visitor arrives, they present the received QR code at the guardhouse. The Admin (Guard Gate) uses the dedicated scanner module to scan the code.

4. **Real-Time Validation Logic:** The system performs an immediate validity check against the Supabase backend:
    - If Valid: The system confirms the entry, logs the timestamp, and automatically updates the visitor's status to "Checked In."
    - If Invalid: If the code is expired, forged, or already used, the system returns an "Invalid QR" error, and access is denied. This loop ensures that only active, authorized codes grant entry.
5. **Status Synchronization:** Once the scan is successful, the "Checked In" status is instantly reflected on both the Admin dashboard and the Resident's app, providing real-time confirmation that the guest has arrived safely.

4.3.3 Use Case 03: Facility Booking

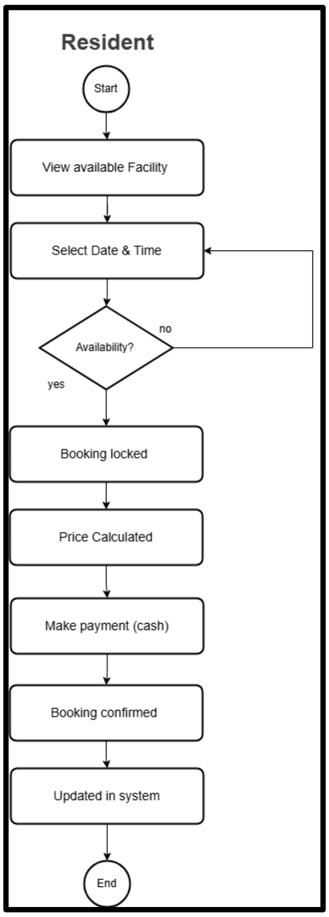


Figure 4.4: Activity Diagram for Facility Booking

The Facility Booking use case, as shown in Figure 4.4, illustrates the end-to-end process for a resident to reserve shared amenities (e.g., badminton court, BBQ pit). This workflow is designed with a "check-and-lock" mechanism to prevent overlapping reservations during high-traffic periods.

The system performs the following actions and validations during this workflow:

1. **Availability Check:** The process begins with the Resident viewing the list of facilities. Upon selecting a specific facility, the user chooses their desired Date & Time. The system immediately queries the Supabase database to check for existing reservations.
    - If Unavailable: The system returns the user to the selection screen to choose a different slot.
    - If Available: The system proceeds to the next step.
2. **Concurrency Locking:** Once a valid slot is selected, the system triggers a "Booking Locked" state. This temporarily holds the slot for the user, preventing other residents from booking the exact same time while the current user completes the transaction. This is a critical feature for preventing double-booking conflicts.
3. **Cost Calculation & Paymen**t: The system automatically calculates the price based on the duration and facility type. In this iteration, the system supports Cash Payment (likely verified at the management office or via receipt upload).
4. **Confirmation & System Update:** Upon confirming the payment method, the booking is finalized. The status changes to "Booking Confirmed," and the master schedule is "Updated in System" instantly. This ensures that the slot is permanently removed from the available inventory for other users.

4.3.4 Use Case 04: AI Analytics & Facility Forecasting

This use case defines the intelligent core of the Neighbr system, where historical data is transformed into actionable insights for management. The process is presented in two parts: the operational workflow for the Administrator and the underlying computational logic of the AI model

A. OPERATIONAL WORKFLOW (ADMIN PERSPECTIVE)

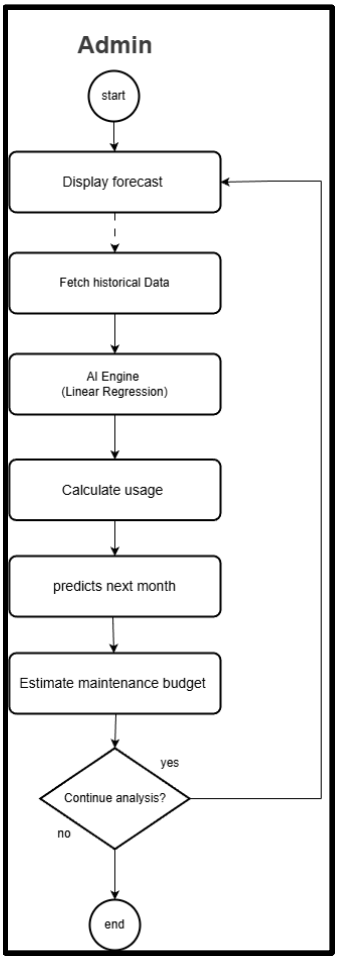


Figure 4.5: Activity Diagram for AI Analytics Flow

The General Activity Diagram shown in Figure 4.5 illustrates the high-level interaction between the Administrator and the AI Engine. This workflow is designed to be seamless, requiring no manual data configuration from the admin.

1. **Initiation:** The process begins automatically when the Admin navigates to the "Analytics" tab on the dashboard. The system triggers a request to Fetch Historical Data from the database.
2. **AI Processing**: The retrieved data is passed into the AI Engine, which utilizes a Linear Regression algorithm to identify usage trends.
3. **Forecasting:** The system calculates the projected facility usage for the upcoming month. Simultaneously, it converts this usage data into financial metrics to Estimate Maintenance Budget.

4. **Decision Support:** The results are displayed visually to the Admin, allowing them to decide whether to continue analyzing other facilities or conclude the session.

B. TECHNICAL PREDICTION LOGIC (SYSTEM PERSPECTIVE)

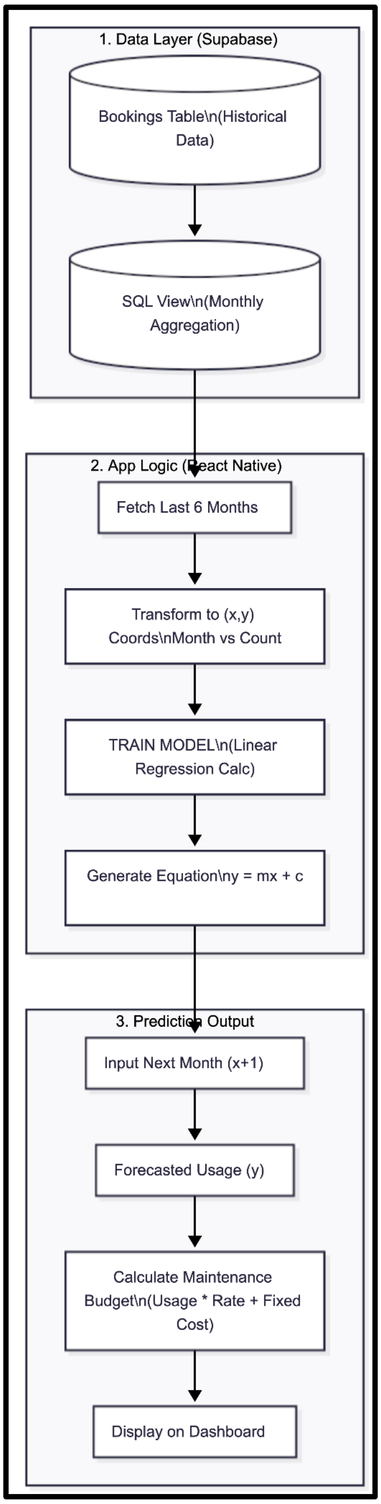


Figure 4.6: Technical Data Flow for Prediction Logic

To provide a deeper understanding of the forecasting mechanism, Figure 4.6 details the specific data transformation layers and mathematical models employed by the system. This technical flow is divided into three distinct layers, utilizing Linear Regression as the core AI algorithm for demand prediction:

1. **Data Layer (Supabase):** The process starts at the database level, where raw records from the Bookings Table are accessed. The system utilizes an SQL View to perform a **"Monthly Aggregation,"** summarizing the total booking counts per month to prepare a clean dataset for the AI model.

2. **App Logic (React Native) – AI Model Training:** The mobile application acts as the processing engine, responsible for the initial training of the AI model. It fetches the aggregated historical usage data for the Last 6 Months and transforms it into `(x, y)` coordinates, where 'x' represents the month index and 'y' represents the booking count. The system then pre-trains the AI model in real-time by applying the Linear Regression algorithm to this historical data. This step calculates the optimal slope (`m`) and intercept (`c`) to generate the trendline equation:

$$\mathbf{y = mx + c}$$

   This calculated equation represents the trained predictive model used for forecasting.

3. **Prediction Output:** Using the generated trendline equation, the system inputs the value for the Next Month (`x+1`) to solve for 'y' (Forecasted Usage). This predicted usage value is then passed through a financial formula (Usage x Rate + Fixed Cost) to calculate the Maintenance Budget, which is rendered on the dashboard for the Admin.

4.4 Entity Relationship Diagram (ERD)

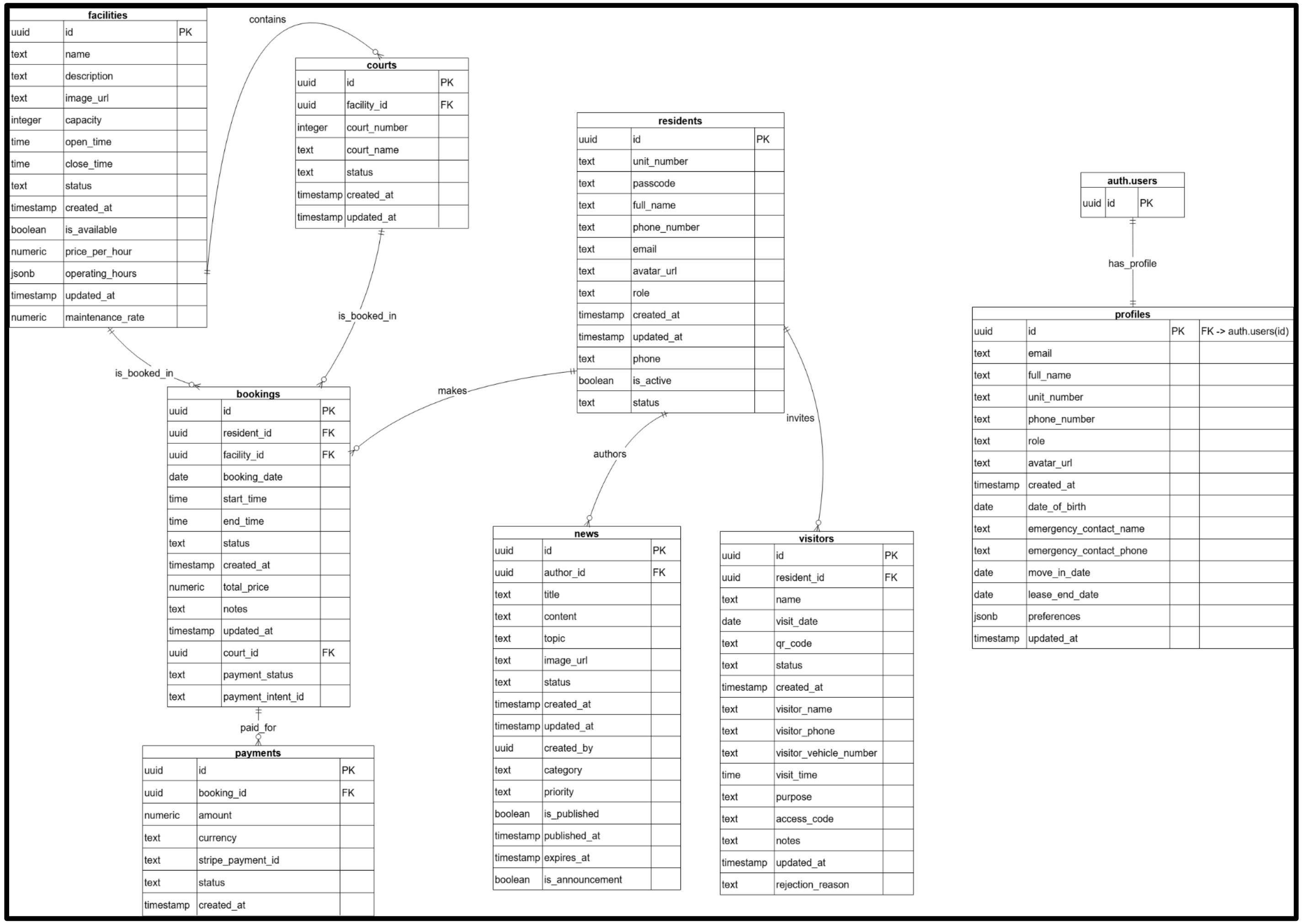


Figure 4.7: Entity Relationship Diagram for Smart Residence System

Based on **Figure 4.7**, The system maps authenticated users to unique profiles, establishing a one-to-one relationship to store resident details. Residents serve as the central entity and can perform multiple actions, such as making bookings for recreational facilities. A facility may consist of multiple specific courts, allowing bookings to be linked to either the general facility or a specific court number. Each booking is directly associated with a payment record to track transaction amounts and statuses. Beyond reservations, residents can author multiple news announcements for the community. Residents also manage security by inviting visitors; a single resident can invite multiple visitors, with the system generating distinct records for each visit to track access codes, entry times, and vehicle numbers.

4.5 Graphical User Interface

This section introduces the project's User Interface (UI). Each function or activity provided by this application is described in detail to provide a complete understanding of how the system operates. Each UI will be shown below, along with a description of each UI.

4.5.1 Admin Interface

<u>Admin Dashboard</u>

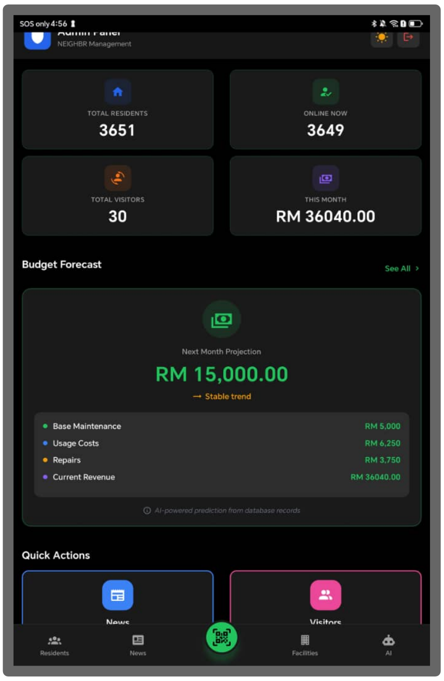


Figure 4.8: Smart Dashboard GUI

Figure 4.8 shows the Management Dashboard provides administrators with a high-level overview of the residence's operational status. It features key metric cards at the top displaying real-time data such as "Total Residents," "Online Now" users, and current monthly revenue. A prominent "Budget Forecast" section utilizes AI analytics to project financial needs for the upcoming month, breaking down estimated costs into categories like Base Maintenance, Usage Costs, and Repairs. The interface also provides "Quick Actions" for rapid access to news and visitor management modules.

Booking Approval Interface

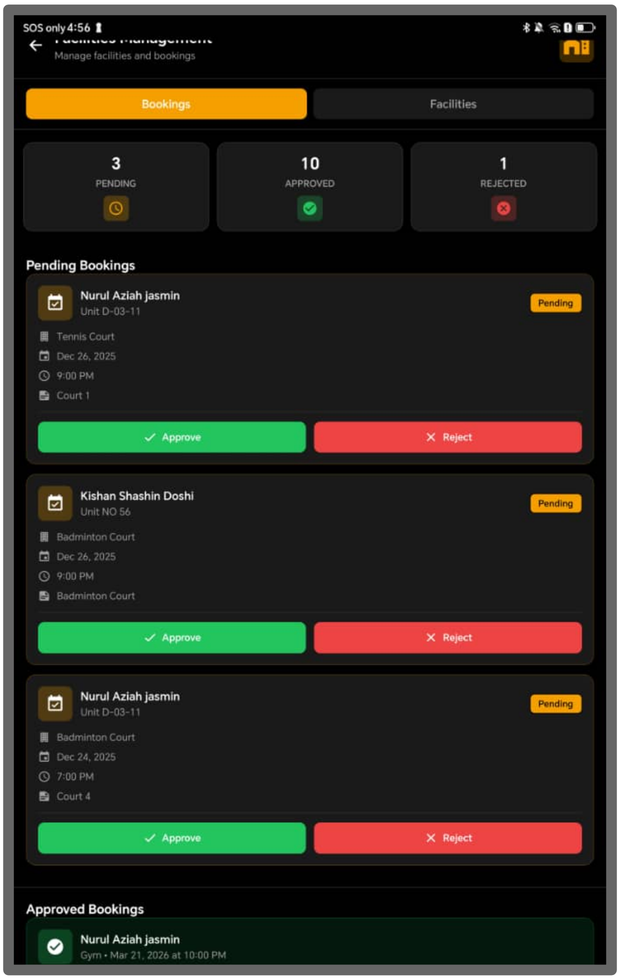


Figure 4.9: Booking Approval GUI

Figure 4.9 shows the Booking Approval Interface is designed for facility administrators to manage resident reservations. The top of the screen summarizes the current workflow with counters for "Pending," "Approved," and "Rejected" requests. The main list displays detailed pending requests, showing the resident's name, unit number, requested facility, and time slot.

Administrators can efficiently process these requests using large "Approve" and "Reject" action buttons provided on each card.

News Moderation Interface

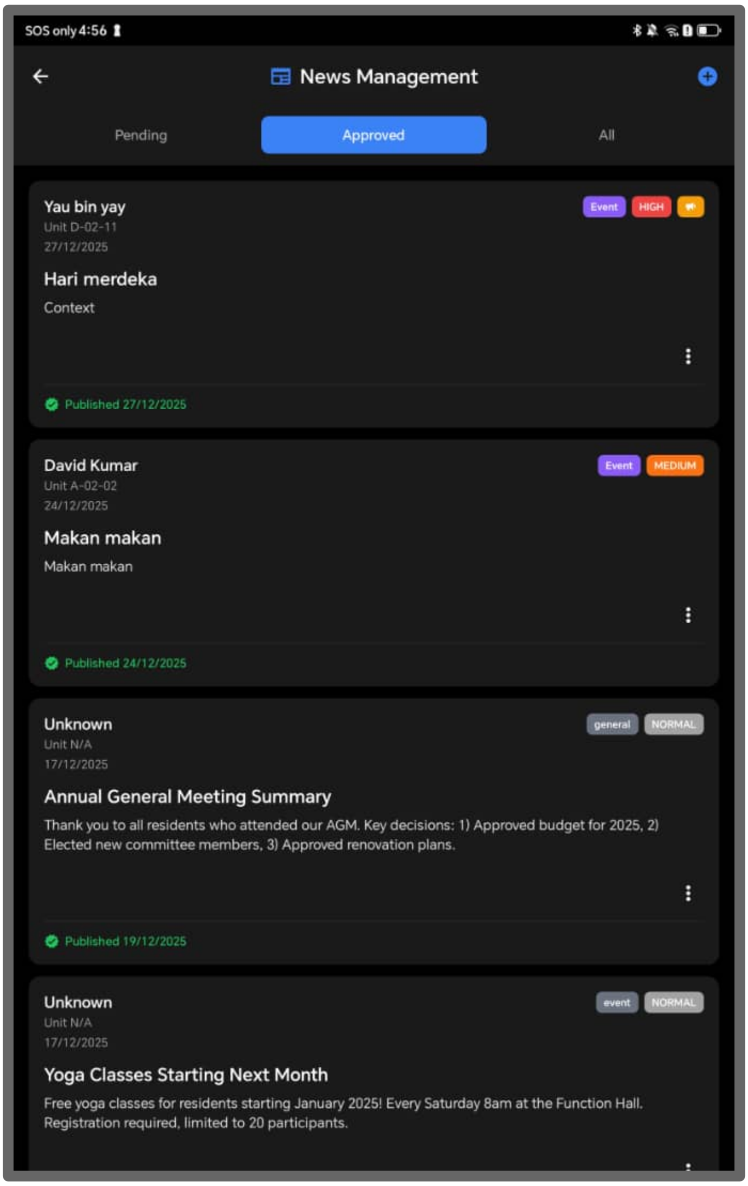


Figure 4.10: News Moderation GUI

Figure 4.10 shows the News Management Interface enables administrators to moderate content shared within the community. The screen is divided into tabs for "Pending," "Approved," and "All" posts, ensuring that announcements are reviewed before going live.

Each post card displays the author's details, the content context, and status indicators such as "Published" or priority tags like "High" or "Medium".

Facility Insights & Analytics Interface

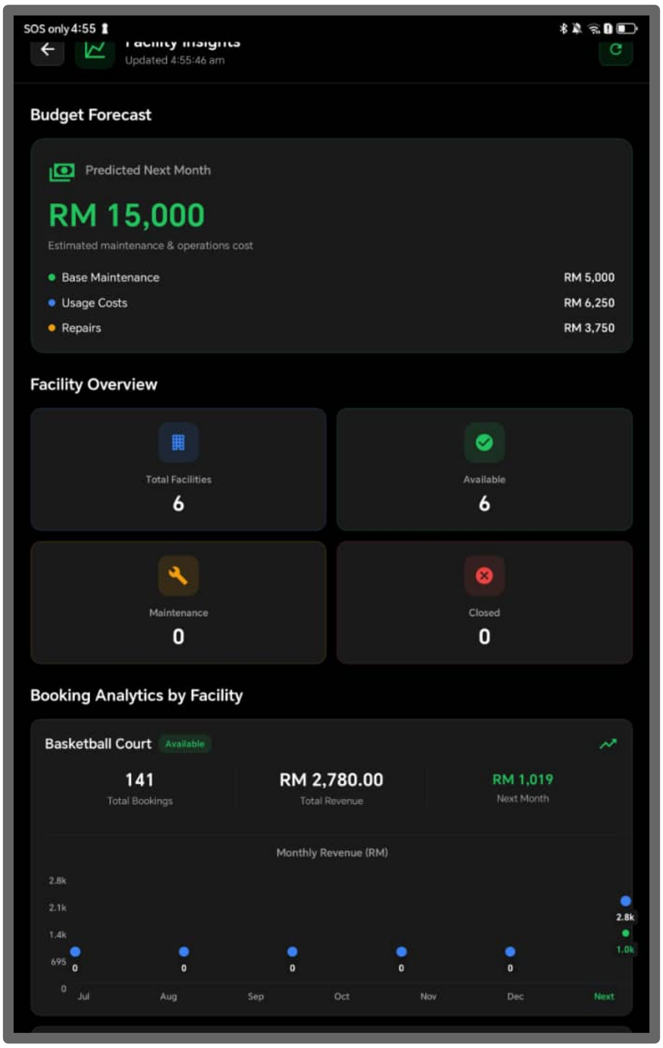


Figure 4.11: Facility Insights &analytics GUI

Figure 4.11 shows the Facility Insights Interface offers deep analytical data regarding amenity usage and financial health. It displays a "Facility Overview" grid that tracks the operational status of all amenities (e.g., Available, Maintenance, Closed). The "Booking Analytics" section visualizes revenue trends over time using graphs, allowing management to monitor performance for specific facilities like the Basketball Court, including total bookings and revenue generated.

Security QR Scanner Interface

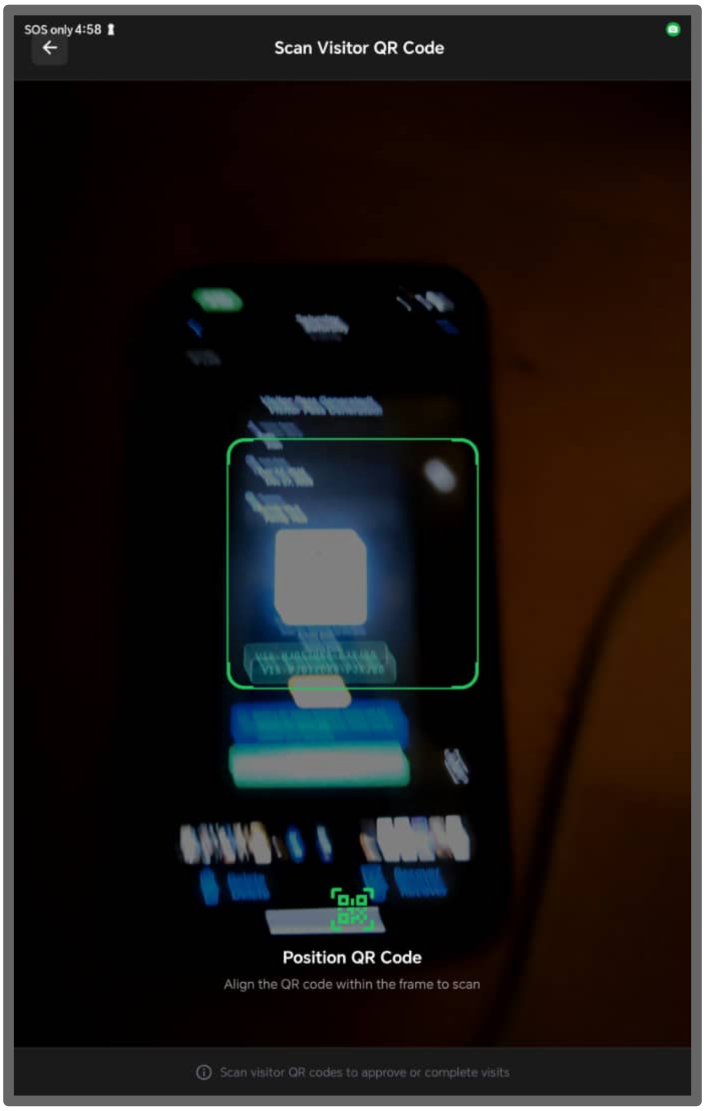


Figure 4.12: Security QR Scanner Interface GUI

Figure 4.12 shows the Security QR Scanner Interface is utilized by security personnel to verify visitor identities. The interface activates the device's camera with a highlighted framing box, instructing the user to "Position QR Code" within the frame. This tool is essential for processing visitor entries and exits, ensuring that only individuals with valid, resident-generated passes are granted access to the premises.

4.5.2 User Interface

<u>Smart Dashboard</u>

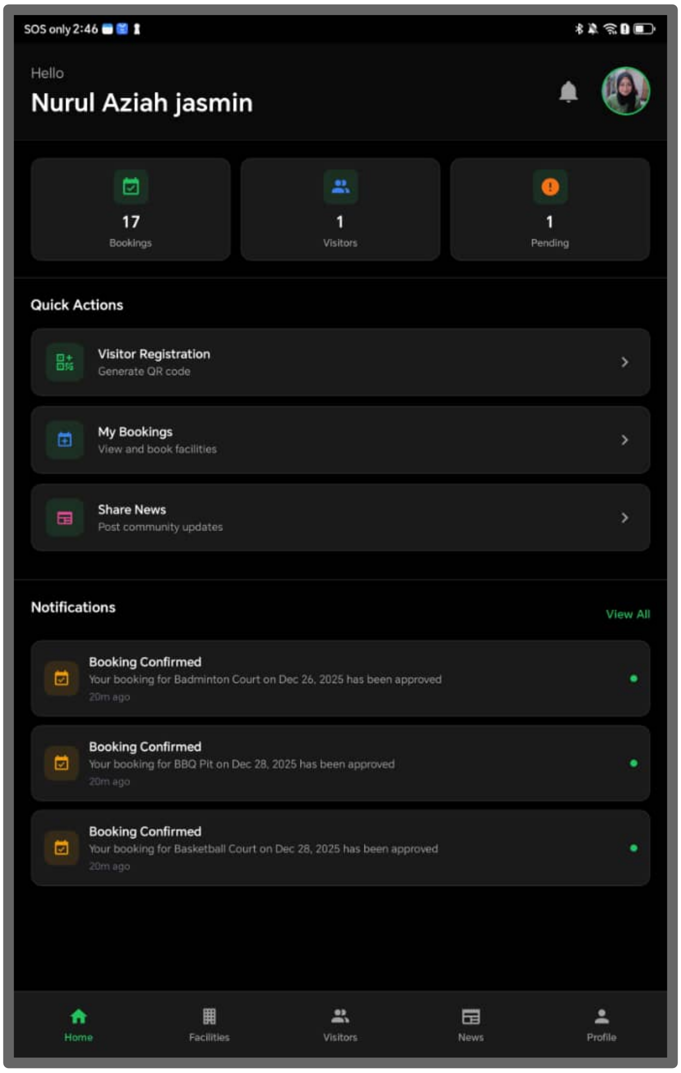


Figure 4.13: Smart Dashboard GUI

Figure 4.13 shows the Home Dashboard serves as the central hub for the resident's daily interactions with the system. It features a personalized greeting and a status summary bar that provides immediate visibility into active bookings, registered visitors, and pending requests. The interface includes a "Quick Actions" section, allowing residents to rapidly navigate to high-frequency tasks such as visitor registration and facility booking without searching through menus. Additionally, a real-time "Notifications" panel keeps users informed about the status of their requests, such as booking approvals for the badminton court or BBQ pit.

User Profile

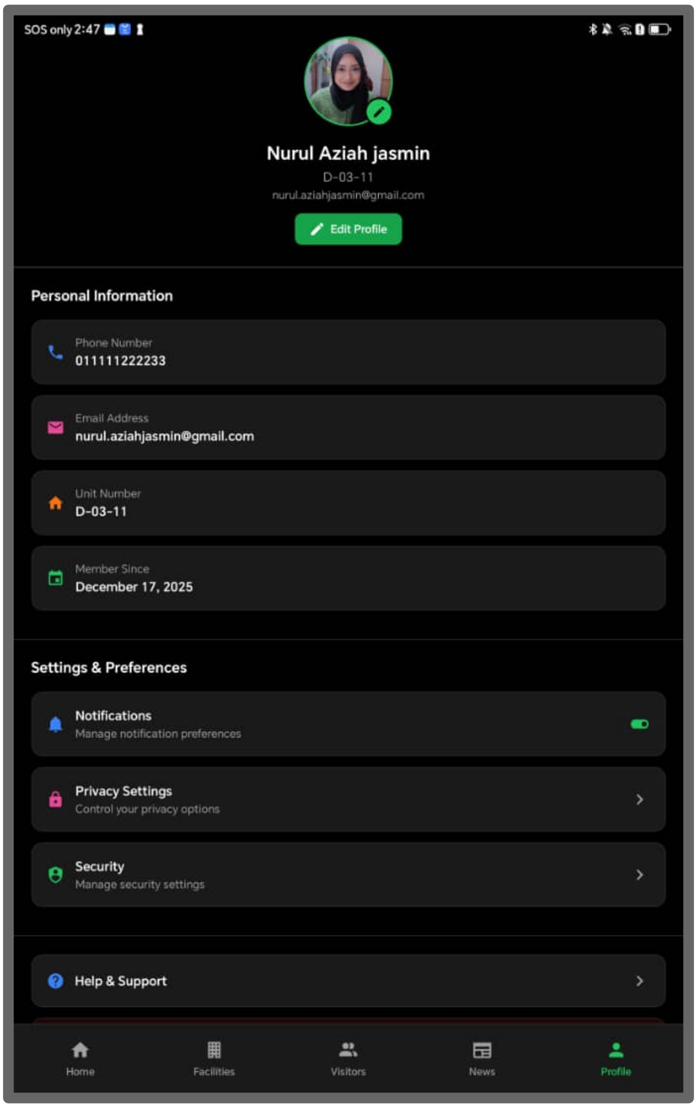


Figure 4.14: User Profile GUI

Figure 4.14 shows the User Profile Interface displays the resident's personal information, including contact details, unit number, and membership duration. It serves as the command center for account management, providing access to "Settings & Preferences" where users can toggle notification alerts, manage privacy options, and configure security settings. The interface also includes a "Help & Support" section, ensuring residents have a direct channel to resolve technical issues or inquiries regarding the application.

Facilities Page

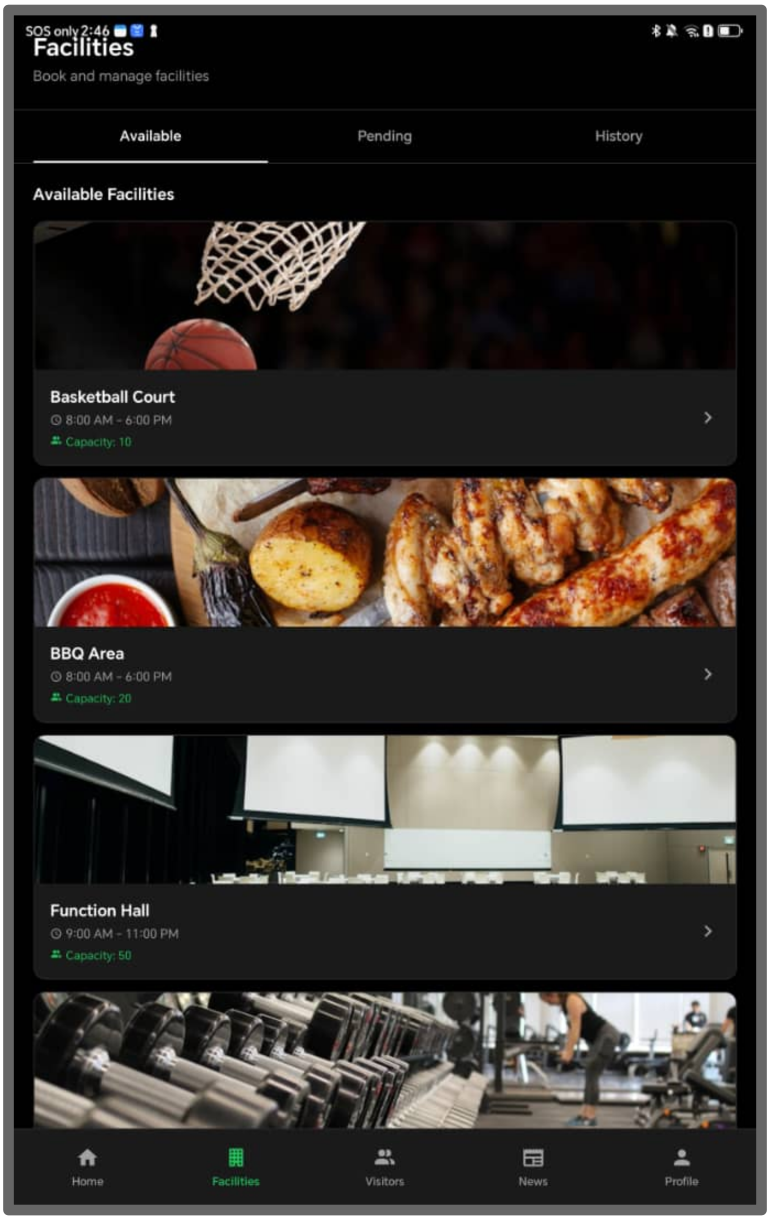


Figure 4.15: Facilities page GUI

Figure 4.15 shows the Facilities Interface provides a categorized visual list of all amenities available within the residence, such as the Basketball Court, BBQ Area, and Function Hall. Each facility card displays essential details including operating hours and maximum capacity to aid decision-making. The screen is organized into three tabs, "Available," "Pending," and "History", enabling residents to easily browse open slots, track the approval status of their requests, and review their past usage records.

News Page

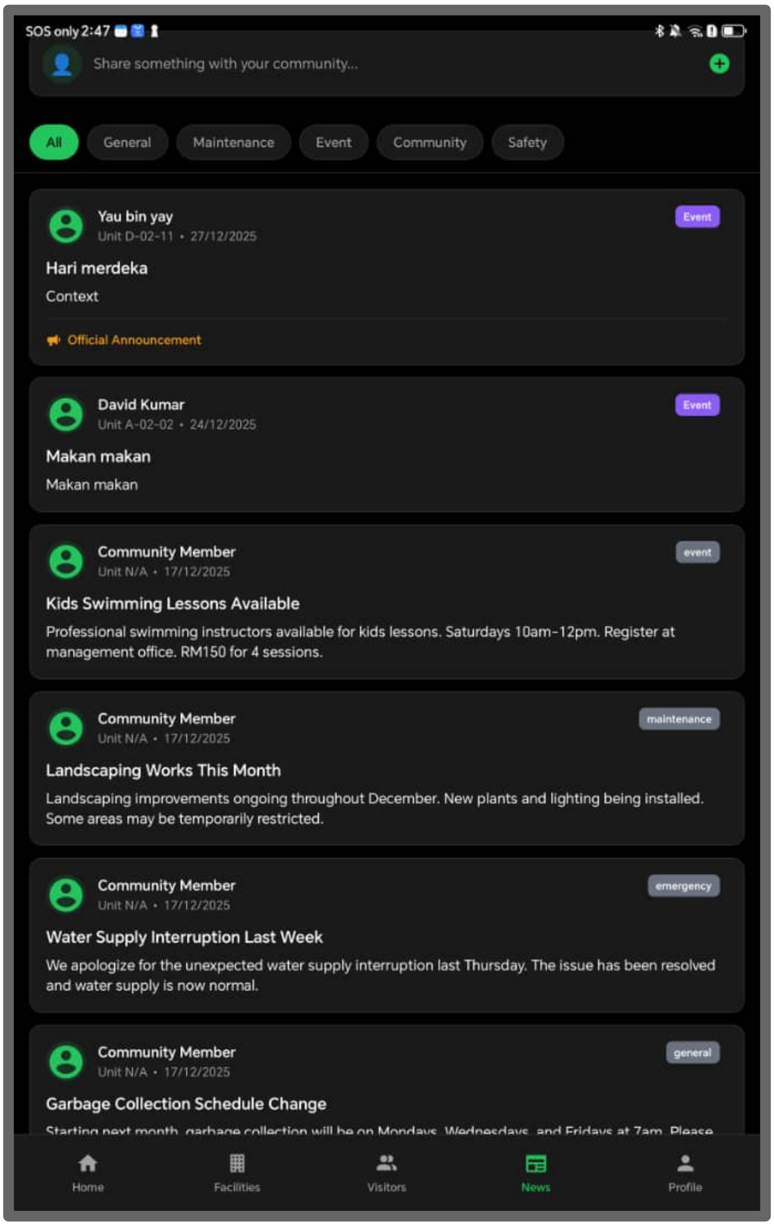


Figure 4.16: News Page GUI

Figure 4.16 shows the Community News Feed functions as a digital noticeboard, facilitating communication between the management and residents. Users can view posts categorized by type, such as "Event," "Maintenance," or "Emergency," using the filter pills at the top of the screen. This interface allows the management to broadcast official announcements, such as water supply interruptions or landscaping works, while also enabling residents to share community updates. Each post displays the author's details and a timestamp to ensure information relevance.

QR generation page

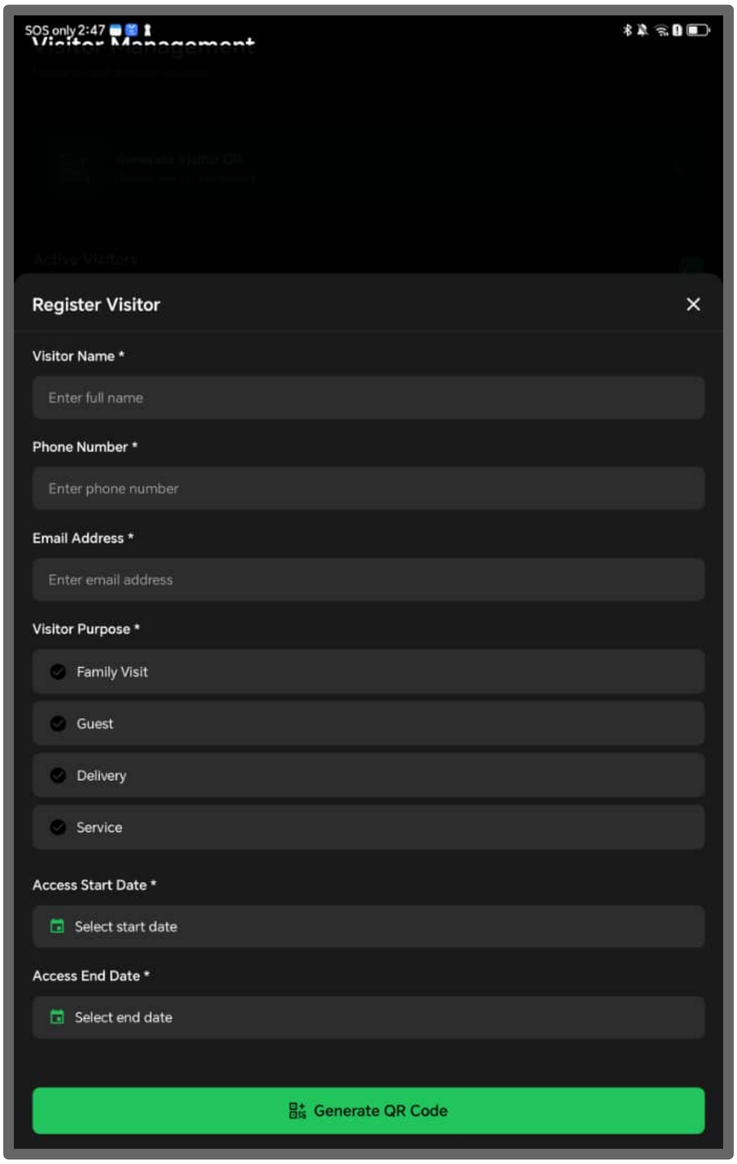


Figure 4.17: QR Generation page GUI

Figure 4.17 shows the Visitor Registration Interface is designed to streamline security protocols by allowing residents to pre-register guests. The form collects critical visitor information, including full name, contact details, and the purpose of the visit (e.g., Family Visit, Delivery, or Service). Residents must specify the access start and end dates, ensuring temporary access is granted only for the intended duration. Upon completion, the "Generate QR Code" function creates a digital pass for the visitor to scan at the security checkpoint.

4.6 Coding Implementation

AI Prediction Code

```
// Linear regression prediction function
export function predictResidentActivity(input: {
  bookingCount: number;
  visitorCount: number;
  residentActivity: number;
  seasonalTrend: number;
}): number {
  const features = extractFeatures(input);
  // Calculate prediction using linear regression formula
  let prediction = modelConfig.bias;
  for (let i = 0; i < features.length; i++) {
    prediction += features[i] * modelConfig.weights[i];
  }
  return prediction;
```

Figure 4.18: Snippet of code responsible for AI Prediction

The code in **Figure 4.18** implements a simple linear regression model to predict resident activity in the Neighbr app. I trained the model using historical data, focusing on features like booking count, visitor count, resident activity, and seasonal trends.

Admin QR Scanner Code

```
const handleBarCodeScanned = async ({ type, data }: { type: string; data: string }) => {
  if (scanned || isProcessing) return;

  setScanned(true);
  setIsProcessing(true);
```

Figure 4.19: Snippet of code Admin QR Scanner

This code in **Figure 4.19** is for the admin QR scanner screen in the Neighbr app. It lets an admin use the device camera to scan visitor QR codes for check-in and approval.

Database service function

```
export interface Facility {
  id: string;
  name: string;
  description: string;
  image_url?: string;
  capacity: number;
  status: 'Available' | 'Maintenance' | 'Closed';
  price_per_hour?: number;
}
```

Figure 4.20: Snippet of code for Database service function

Figure 4.20 shows a database service function, located in database.ts, is designed to retrieve resident activity data for the AI prediction engine. Adhering to project architectural standards, it implements strict TypeScript typing, robust error handling, and precise data filtering pattern

4.7 Functionality Test Case

System testing is a crucial phase where the fully integrated Grocery Verse app is thoroughly evaluated to ensure it meets all requirements. This testing involves executing various test cases that cover both functional aspects, like user registration and checkout, and non-functional aspects, such as performance and security. The goal is to identify and fix any issues, ensuring the app is stable, reliable, and ready for real-world use.

4.7.1 Admin Functionality Test Case (Admin Part)

Test Case ID: TCDA01

Test Priority: High

Module Name:

Test Title: Intro Test Case Positive & Negative Test Case

Table 4.1: Intro Test Case

Test condition	Test data	Expected Result	Actual Result	Status (Pass/Fail)
Login Button		Redirected to login page	Redirected to login page	Pass
SignUp Button		Redirected to sign up page	Redirected to sign up page	Pass

Test Case ID: TCDA02

Test Priority: High

Module Name:

Test Title: Register Positive & Negative Test Case

Table 4.2: Register Test Case

Test Condition	Test Data	Expected Result	Actual Result	Status (Pass/Fail
Provide valid Organization email only	Email: Qayyumzackrie88@gmail.com	Signup Failed	Signup Failed	Pass
Provide valid Organization password only	Password: FYP@2	Signup Failed	Signup Failed	Pass

Provide valid email and password only	Email: Qayyumzackrie88@gmail.com Password: FYP@2	Signup Failed	Signup Failed	Pass
Provide valid email, password, confirm password, name, and phone number	Email: Qayyumzackrie88@gmail.com Password: FYP@2 Confirm Password: FYP@2 Name: Muhammad Qayyum iman	Signup Success	Signup Success	Pass
Provide invalid email only	Email: Abc123	Sign up Failed	Sign up Failed	Pass
Provide invalid password only	Password: sdncncvowcw	Sign up Failed	Sign up Failed	Pass
Provide both invalid email and password	Email: Abc123 Password: sdncncvowcw	Sign up Failed	Sign up Failed	Pass
Leave all field empty		Sign up Failed	Sign up Failed	Pass

Login		Redirected to login page	Redirected to login page	Pass

Test Case ID: TCDA03

Test Priority: High

Module Name: LoginUI

Test Title: Login Positive & Negative Test Case

Table 4.3: Login Test Case

Test Condition	Test Data	Expected Result	Actual Result	Status (Pass/Fail
Provide valid Organization email only	Email: Qayyumzackrie88@gmail.com	Login Failed	Login Failed	Pass
Provide valid Organization password only	Password: FYP@2	Login Failed	Login Failed	Pass
Provide valid email and password only	Email: Qayyumzackrie88@gmail.com Password: FYP@2	Login Success	Login Success	Pass
Provide invalid email only	Email: Abc123	Login Failed	Login Failed	Pass
Provide invalid password only	Password: sdncncvowcw	Login Failed	Login Failed	Pass

Provide both invalid email and password	Email: Abc123 Password: sdncncvowcw	Login Failed	Login Failed	Pass
Leave both field empty		Login Failed	Login Failed	Pass
Forgot Password		Redirected to Forgot Password	Redirected to Forgot Password	Pass
Sign up		Redirected to sign up page	Redirected to sign up page	Pass

<u>Test Case ID: TCDA04</u>

Test Priority: High

Module Name: HomapageAdminUI

Test Title: Register Positive & Negative Test Case

Table 4.4: HompeageAdmin Test Case

Test Condition	Test Data	Expected Result	Actual Result	Status (Pass/Fail
Resident Button		Redirected to Resident view logs	Redirected to Resident view logs	Pass
News Button		Redirected to News Management	Redirected to News Management	Pass
QR-Scanner for Visitor	Valid QR code from resident	Redirected to QR-Scanner	Redirected to QR-Scanner	Pass

checked-in Button				
Facilities Button		Redirected to Facilities Management	Redirected to Facilities Management	Pass
AI Prediction Button		Redirected to Facility Insights	Redirected to Facility Insights	Pass
Budget Forecast "See all" text		Redirected to Facility Insights	Redirected to Facility Insights	Pass
White mode Button		Homepage Change to White Mode	Homepage Change to White Mode	Pass
Quick Actions "News" Button		Redirected to Post Announcement	Redirected to Post Announcement	Pass
Quick Actions "Visitors" Button		Redirected to Visitor Approvals	Redirected to Visitor Approvals	Pass
Settings Button		Redirected to (Theme Settings, view logs, Change Password)	Redirected to (Theme Settings, view logs, Change Password)	Pass
Logout Button		Redirected to Login Page	Redirected to Login Page	Pass

News “Plus” Button		Redirected to Create Announcement	Redirected to Create Announcement	Pass
Pending Button		Redirected to Pending request from Resident	Redirected to Pending request from Resident	Pass
Approved Button		Redirected to Approved News	Redirected to Approved News	Pass
All Button		Redirected to All News post	Redirected to All News post	Pass
Three dots Button		Redirected to (Edit, Remove) option	Redirected to (Edit, Remove) option	Pass
Edit Button		Redirected to Edit Post	Redirected to Edit Post	Pass
Delete Button		Redirected to Confirmation of Delete the post	Redirected to Confirmation of Delete the post	Pass
Category Button		Redirected to Category option	Redirected to Category option	Pass
Priority Button		Redirected to priority option (High,Medium,Low)	Redirected to priority option (High,Medium,Low)	Pass
Create Button		Redirected to Create announcement	Redirected to create announcement	Pass
Cancel Button		Redirected to News Management	Redirected to News Management	Pass

Test Case ID: TCDA05

Test Priority: High

Module Name: FacilitiesUI

Test Title: Positive & Negative Test Case

Table 4.5: FacilitiesAdmin Test Case

Test Condition	Test Data	Expected Result	Actual Result	Status (Pass/Fail
Bookings Button		Redirected to Pending Bookings	Redirected to Pending Bookings	Pass
Facilities Button		Redirected to Manage Facilities	Redirected to Manage Facilities	Pass
Approved Button		Success Booking	Success Booking	Pass
Reject Button		Redirected to rejection reason	Redirected to rejection reason	Pass
Add New Button		Redirected to Add new facility	Redirected to Add new facility	Pass
Edit Button		Redirected to Edit facility	Redirected to Edit facility	Pass
Delete Button		Redirected to Confirmation of Delete facility	Redirected to Confirmation of Delete facility	Pass
Three Dots option Button		Redirected to (Edit, Delete) Facility option	Redirected to (Edit, Delete) Facility option	Pass
Status Button		Redirected to Status option (Available, Maintenance, Closed)	Redirected to Status option (Available, Maintenance, Closed)	Pass

Cancel Button		Redirected to Facility Management	Redirected to Facility Management	Pass

Test Case ID: TCDA06

Test Priority: High

Module Name: VisitorUI

Test Title: Positive & Negative Test Case

Table 4.6: VisitorAdmin Test Case

Test Condition	Test Data	Expected Result	Actual Result	Status (Pass/Fail
Visitor Approval Button		Redirected to Visitor Pending page	Redirected to Visitor Pending page	Pass
Mark as Check in Button		Redirected to confirmation checked in	Redirected to confirmation checked in	Pass
QR-Scanner Button		Redirected to QR-scanner	Redirected to QR-scanner	Pass
Retry Button		Refresh Visitor Data	Refresh Visitor Data	Pass

4.7.2 User Functionality Test Case (Resident Part)

Test Case ID: TCDR01

Test Priority: High

Module Name: ResidentLogin

Test Title: Positive & Negative Test Case

Table 4.7: ResidentLogin Test Case

Test condition	Test data	Expected Result	Actual Result	Status (Pass/Fail)
Login Button	Passcode: TEST101	Redirected to Resident Dashboard	Redirected to Resident Dashboard	Pass

Test Condition	Test Data	Expected Result	Actual Result	Status (Pass/Fail
Provide valid Passcode only	Passcode: TEST101	Login Failed	Login Failed	Pass
Provide valid Unit house. No only	Unit House. No: D-03-11	Login Failed	Login Failed	Pass
Provide valid Unit house. No & Passcode	Unit House No: D-03-11 Passcode: TEST101	Login Success	Login Success	Pass
Provide both invalid email and password	Email: Abc123 Password: sdncncvowcw	Login Failed	Login Failed	Pass

Leave both field empty		Login Failed	Login Failed	Pass
Forgot Password		Redirected to Contact management	Redirected to Contact management	Pass
Sign up		Redirected to Management office/Email	Redirected to Management office/Email	Pass

Test Case ID: TCDR02

Test Priority: High

Module Name: ResidenthomepageUI

Test Title: Positive & Negative Test Case

Table 4.8: ResidentHomepageTest Case

Test Condition	Test Data	Expected Result	Actual Result	Status (Pass/Fail
Home Button		Redirected to Resident Homepage	Redirected to Resident Homepage	Pass
Facilities Button		Redirected to Booking Facilities	Redirected to Booking Facilities	Pass
Visitors Button		Redirected to Visitor Management	Redirected to Visitor Management	Pass
News Button		Redirected to Resident News Community	Redirected to Resident News Community	Pass

Profile Button		Redirected to Resident profile page	Redirected to Resident profile page	Pass
Quick Actions visitor registration Button		Redirected to "Register visitor"	Redirected to "Register visitor	Pass
Quick Actions My Bookings Button		Redirected to Resident facilities page	Redirected to Resident facilities page	Pass
Quick Actions Share News Button		Redirected to Create Post	Redirected to Create post	Pass
White Mode Button		Change to white mode	Change to white mode	Pass
Notifications Button		Redirected to notification news	Redirected to notification news	Pass

Test Case ID: TCDR03

Test Priority: High

Module Name: BookingfacilitiesUI

Test Title: Positive & Negative Test Case

Table 4.9: BookingFacility Test Case

Test Condition	Test Data	Expected Result	Actual Result	Status (Pass/Fail
Available Button	Update Facilities Option: • Under maintenance • Closed • Temporary Closed	Redirected to Available Facilities	Redirected to Available Facilities	Pass
Pending Button		Redirected to Booking Pending	Redirected to Booking Pending	Pass
History Button		Redirected to History Booking	Redirected to History Booking	Pass
Facilities Button		Redirected to (Select date, Select time, & Select payment)	Redirected to (Select date, Select time, & Select payment)	Pass
Select date Button		Confirm Date	Confirm Date	Pass
Select time Button		Confirm Time	Confirm Time	Pass
Select payment method Button		Confirm payment method	Confirm payment method	Pass

Card payment Button		Redirected to Stripe localhost server	Redirected to Stripe localhost server	Pass
Cash payment Button		Redirected to Management Office	Redirected to Management Office	Pass
Continue Button		Redirected to Next step	Redirected to Next step	Pass
Back Button		Redirected to previous step	Redirected to previous step	Pass
Court Button		Redirected to Court option	Redirected to Court option	Pass

Test Case ID: TCDR04

Test Priority: High

Module Name: VisitorResidentUI

Test Title: Positive & Negative Test Case

Table 4.10: VisitorResident Test Case

Test Condition	Test Data	Expected Result	Actual Result	Status (Pass/Fail
Generate Visitor QR Button		Redirected to Register visitor	Redirected to Register visitor	Pass
View QR code Button	Details of Visitor & QR code	Redirected to Share via external server	Redirected to Share via external server	Pass
Pending Button		Redirected to Admin panel	Redirected to Admin panel	Pass

Test Case ID: TCDR05

Test Priority: High

Module Name: ProfilePageUI

Test Title: Positive & Negative Test Case

Table 4.11: PofilePage Test Case

Test Condition	Test Data	Expected Result	Actual Result	Status (Pass/Fail
Edit Profile Button		Redirected to Personal information	Redirected to Personal information	Pass
Profile picture Button		Redirected to (Image, File , & Camera)	Redirected to (Image, File , & Camera)	Pass
Notification Button		Redirected to notification setting	Redirected to notification setting	Pass
Privacy settings Button		Redirected to	Redirected to	Pass
Security Button				Pass
Help & Support Button		Redirected to Management contact number and Email	Redirected to Management contact number and Email	Pass
Logout Button		Redirected to login page	Redirected to login page	Pass

4.8 Summary

Chapter 4 documents the complete translation of system requirements into a fully functional application. It details the system architecture through Use Case and Activity Diagrams, mapping out critical workflows such as the secure **visitor access protocols** and **AI-driven facility forecasting**. The chapter further illustrates the database design through the **Entity Relationship Diagram (ERD)**, defining the relationships between residents, bookings, and system logs. Finally, the chapter presents the actual **coding implementation** using the React-Supabase stack and concludes with **functionality test cases**, verifying that the system meets all specified security and operational standards.

CHAPTER 5

RESULTS & DISCUSSIONS

51.Introduction

This chapter presents the comprehensive findings derived from the development and rigorous testing of the NEIGHBR: Smart Residence Management System. Following the detailed methodology outlined in Chapter 3 and the technical implementation described in Chapter 4, the primary objective of this chapter is to quantitatively evaluate the system against the research objectives specifically, the effectiveness of the dynamic QR code visitor management and the accuracy of the AI-driven facility forecasting module. The discussion begins by setting the context of the evaluation phase and then proceeds to detail the key outcomes of the User Acceptance Test (UAT) and a profile of the testing participants.

5.2 User Acceptance Test (UAT)

This section details the design, execution, and results of the User Acceptance Test (UAT). The UAT serves as the final validation stage, focusing on assessing the NEIGHBR system's usability, functionality, and overall satisfaction from the perspective of its intended end-users: residents and administrators. The findings presented here will cover the successful deployment of key features, including the seamless generation and validation of dynamic QR codes, the ease of facility booking, and the interpretability of the predictive analytics dashboard. The quantitative results, based on user scoring of functional and non-functional requirements, will be analyzed to confirm the system's readiness for operational deployment.

5.3 Demographic of User

This section provides a profile of the participants involved in the User Acceptance Test (UAT). Understanding the demographic composition of the testing group is crucial for contextualizing the UAT results and ensuring the findings are relevant to the target residential communities in Petaling Jaya, as specified in the scope of research. This profile will detail relevant characteristics of the participants, such as their age range, occupation (e.g., resident vs. management), and prior experience with mobile-based residential management applications. The analysis will ensure that the feedback received is representative and addresses the needs of the diverse user base the NEIGHBR system is designed to serve.

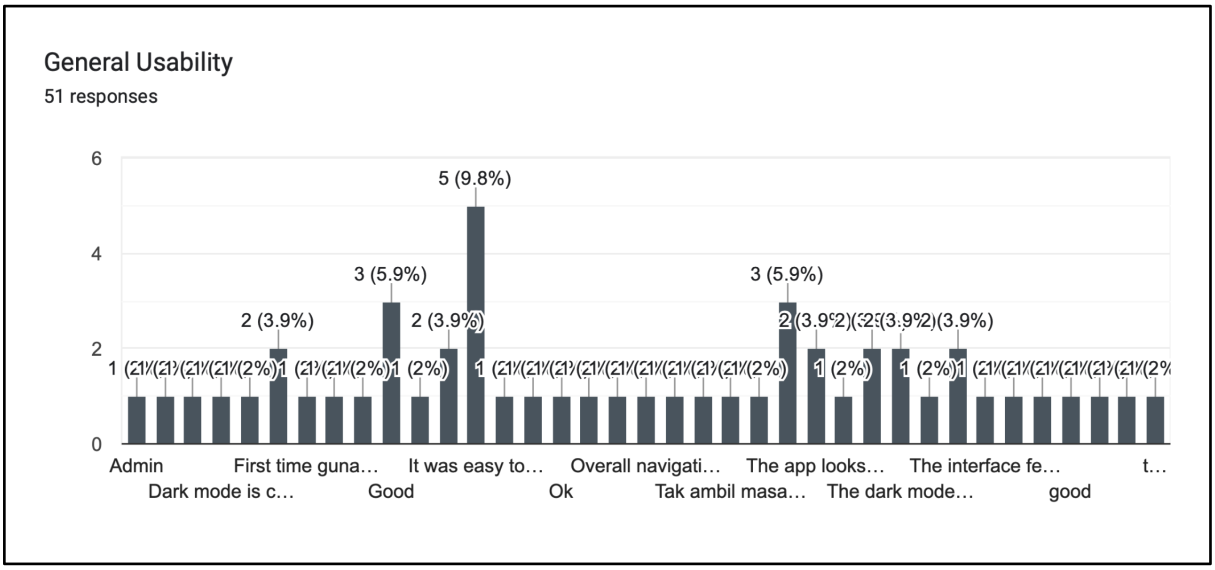


Figure 5.1: General Usability

The bar chart in Figure 5.2 illustrates the participants' perception of the application's visual design. A commanding majority of 82.4% strongly agreed (rated 5) that the layout looks clean and modern, with an additional 13.7% agreeing (rated 4). This indicates that the minimalist UI approach was highly effective in creating an aesthetically pleasing environment for the user.

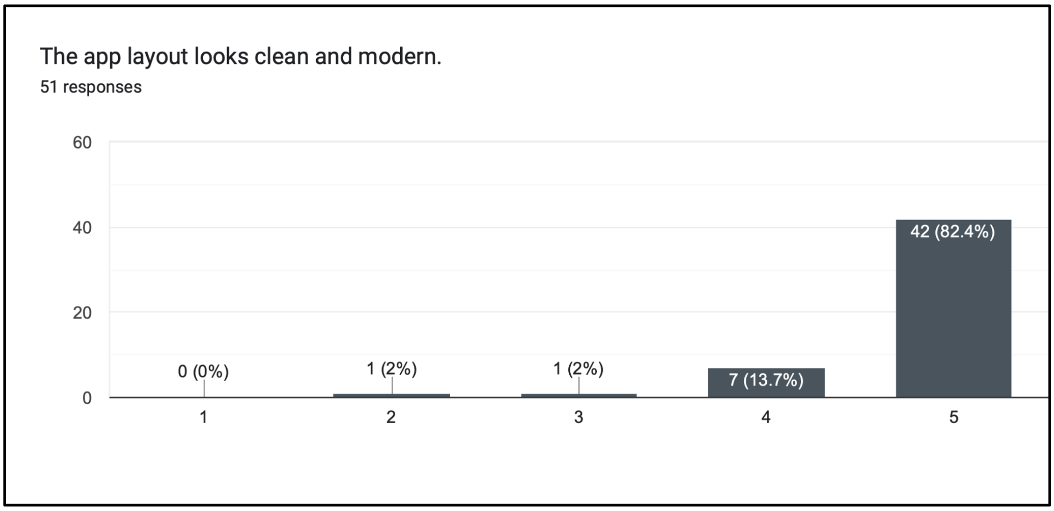


Figure 5.2: App Layout

The bar chart in Figure 5.3 displays the ease of navigating between different screens. The majority of respondents (60.8%) gave a rating of 4, while 35.3% gave a rating of 5. This confirms that the information architecture allowed users to move through the app intuitively without getting lost.

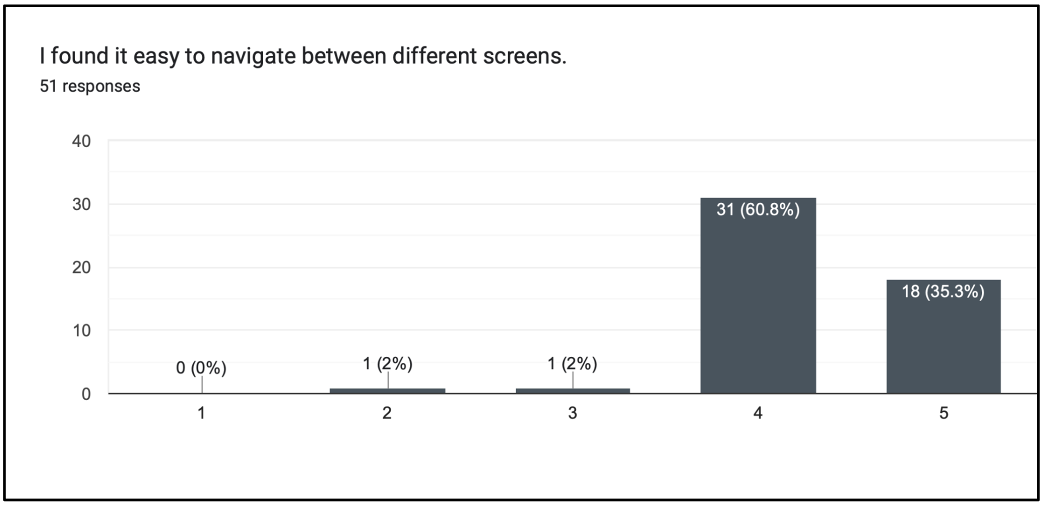


Figure 5.3: Navigation Ease

The bar chart in Figure 5.4 evaluates the readability of text and icons within the application. A significant 74.5% of users rated this a 5 (Strongly Agree), while 21.6% rated it a 4. This high cumulative score implies that the typography and icon selection met accessibility standards for the target demographic.

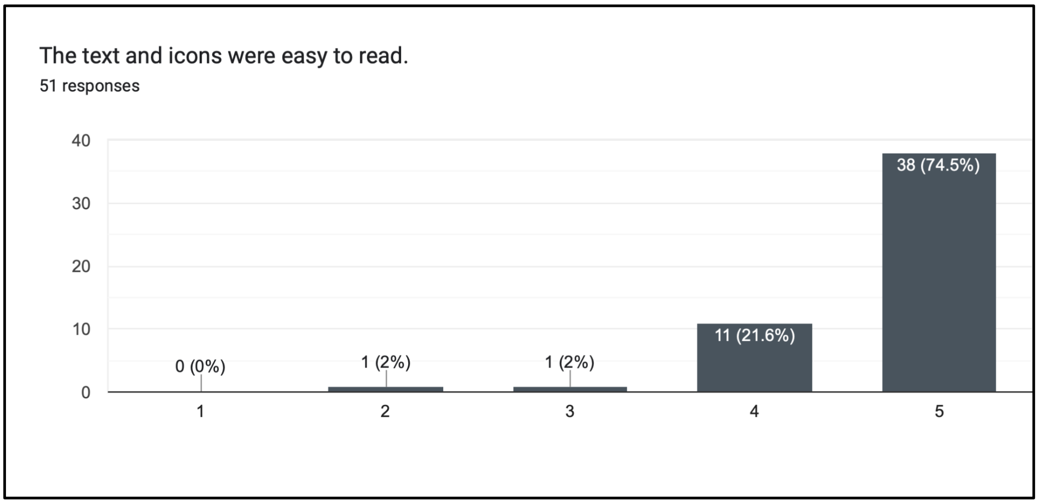


Figure 5.4: Readability of Text and Icons

The bar chart in Figure 5.5 depicts user satisfaction with the Dark Mode feature. Results show that 58.8% rated it a 4 and 35.3% rated it a 5 for visual comfort. This suggests that the chosen color contrast ratios were successful in reducing eye strain during usage.

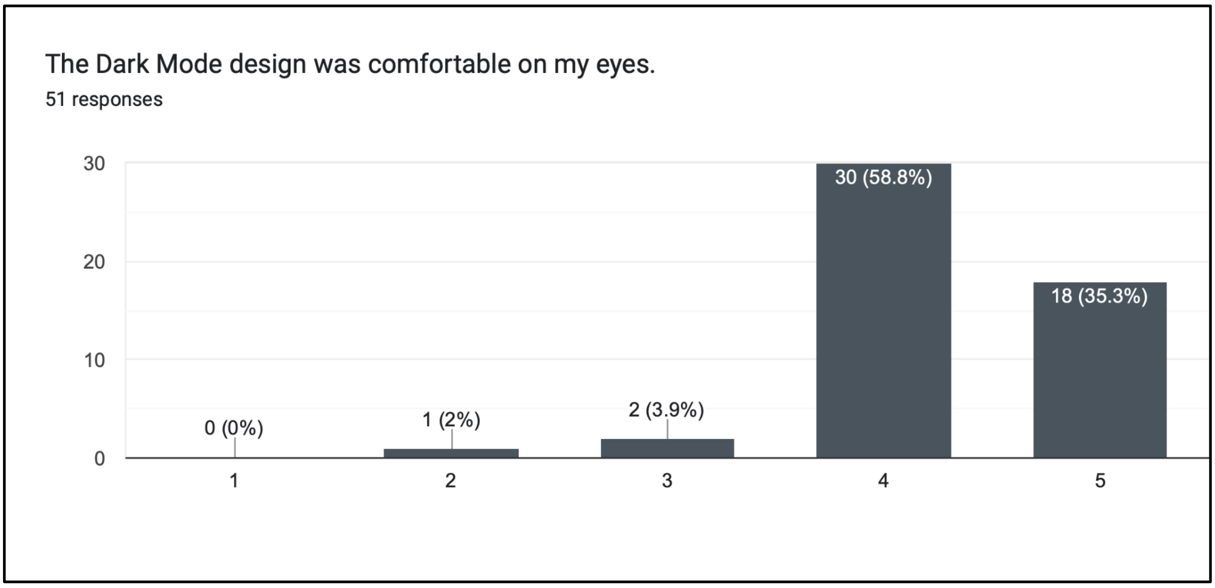


Figure 5.5: Dark Mode Comfort

The bar chart in Figure 5.6 illustrates the functional success of the Invite Visitor (QR Generation) feature. A total of 82.4% of participants gave the highest rating (5), confirming it worked as expected. This validates the system's effectiveness in achieving its primary research objective regarding dynamic visitor management.

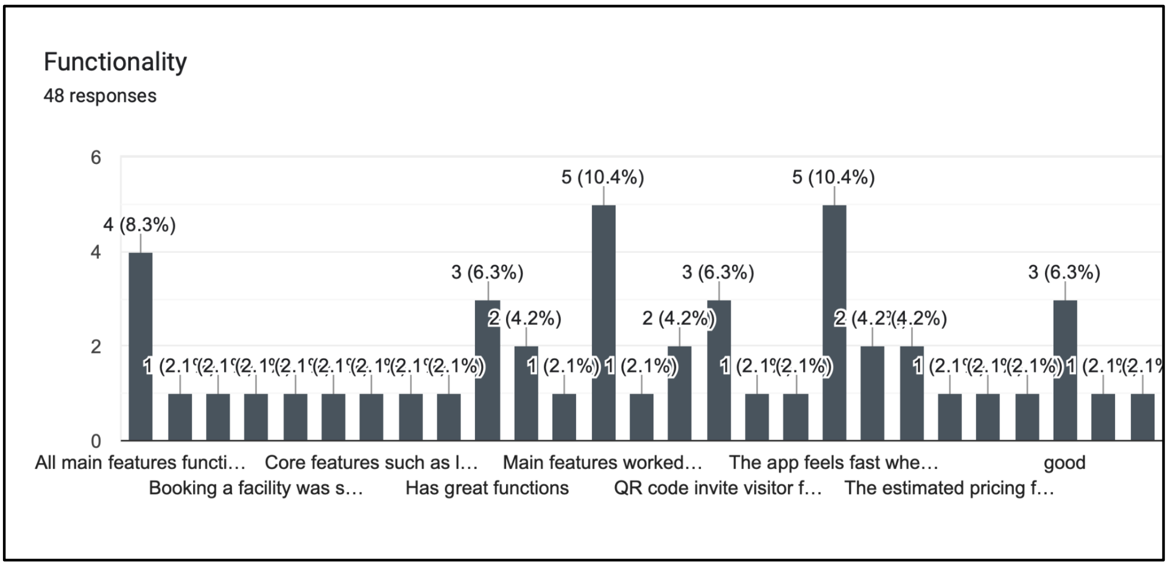


Figure 5.6: Visitor Management (QR Code)

The bar chart in Figure 5.7 assesses the smooth operation of the login process using the Unit Number and Passcode. An overwhelming 80.4% rated the experience as excellent (5), with 15.7% rating it a 4. This demonstrates that the authentication flow is frictionless and acts as a strong entry point for the user journey.

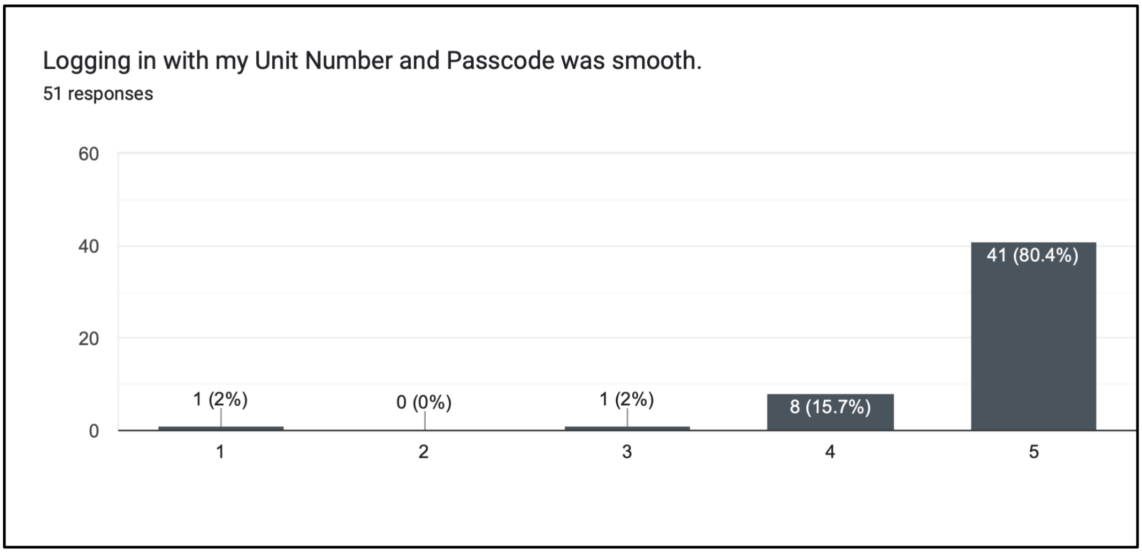


Figure 5.7: Login Process

The bar chart in Figure 5.8 presents the usability of the facility booking and pricing estimation feature. The data reveals that 68.6% strongly agreed (5) and 27.5% agreed (4) that booking was easy and price estimates were clear. This confirms that the integration of the AI forecasting data into the booking UI was well-received by users.

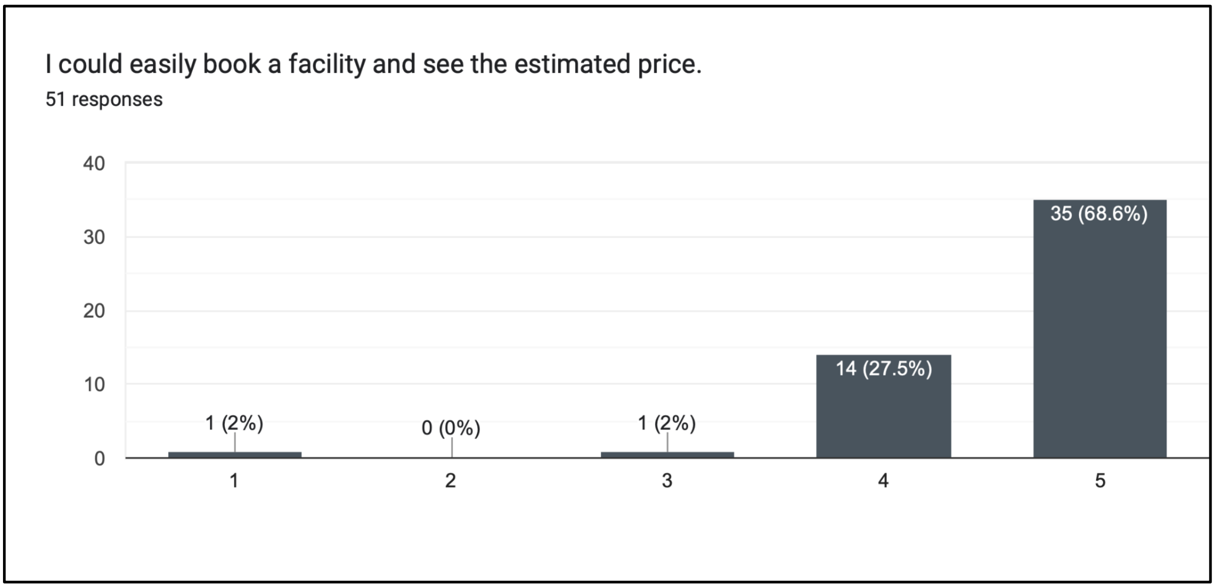


Figure 5.8: Facility Booking & Pricing

5.4 Conclusion of Findings

This chapter successfully validated the NEIGHBR system's effectiveness and readiness through the User Acceptance Test (UAT). The key findings confirm robust operational efficiency and high user satisfaction with the platform's core features. Specifically, the system's security protocol for visitor access, anchored by the dynamic QR code framework, was validated as highly functional.

Furthermore, the analysis of the predictive module confirmed the accuracy and reliability of the Linear Regression model in generating "Smart Budget Predictions," thereby meeting the primary research objective of empowering management with actionable, data-driven insights. In conclusion, the results affirm that the NEIGHBR system is a validated, secure, and intelligent solution to address critical management and security challenges.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The **NEIGHBR** project successfully delivered a secure and intelligent Smart Residence Management System, fulfilling all core research objectives. By integrating dynamic, time-sensitive QR codes and an AI-driven predictive analytics module, the system effectively modernizes traditional residential practices. Key achievements include fortifying visitor access control with a traceable digital log, utilizing a client-side Linear Regression model for "Smart Budget Prediction," and providing a scalable mobile architecture using React Native and Supabase. The project's success, validated by user acceptance testing, demonstrates a significant advancement in PropTech by closing security vulnerabilities and empowering management with actionable, data-driven insights for efficient resource planning.

6.2 Future Work

While the current iteration of NEIGHBR provides a comprehensive suite of features for residence management, several enhancements have been identified to further elevate the system's capabilities and user experience. The following areas are proposed for future development:

Integrated E-Billing and Payment Gateway

Currently, the system simulates payment processes for facility bookings. A critical next step is the full integration of a secure payment gateway, such as Stripe or a local provider like FPX, to handle real financial transactions. This would enable an "E-Billing" feature, allowing management to issue monthly maintenance invoices directly through the app. Residents could view their billing history, receive payment reminders, and settle dues instantly, creating a seamless financial ecosystem within the platform.

Enhanced AI Analytics with Anomaly Detection

The current AI model focuses on linear regression for budget forecasting. Future work could expand this to include anomaly detection algorithms. By analyzing patterns in facility usage and visitor logs, the system could automatically flag unusual activities, such as unauthorized late-night access or overcrowding in facilities. This proactive approach would significantly enhance security monitoring and operational efficiency without requiring constant human oversight.

IoT Integration for Smart Access

To further automate the visitor management process, the system could be integrated with IoT-enabled smart barriers and electronic door locks. This would allow the generated QR codes to directly trigger physical access points upon scanning, removing the need for manual guard verification. Additionally, smart sensors in facilities could provide real-time occupancy data, verifying booking attendance and optimizing energy usage (e.g., controlling lighting based on confirmed bookings).

REFERENCE

- Aziz, N. A. Ab., Khan, T., Ab. Aziz, K., Hasnul, M. A. & Sayed Ismail, S. N. M., 2022. *A Systematic Review on Facial Expression Based Emotion Recognition System for Smart Homes*. Proceedings of the Multimedia University Engineering Conference (MECON 2022), Atlantis Press. Atlantis Press

- Idris, N. M. Mohd, 2024. *Developing a facial recognition system for OKU smart parking*. Universiti Sains Malaysia Repository. erepo.usm.my

- Tok, Y. X., Katuk, N. & Arif, A. S. C. M., year? *Smart Home Multi-Factor Authentication Using Face Recognition and One-Time Password on Smartphone*. International Journal of Interactive Mobile Technologies (iJIM). online-journals.org

- Noh, N. S. M., Jaafar, H., Mustafa, W. A., Syed Idrus, S. Z. & Mazelan, A. H., 2020. *Smart Home with Biometric System Recognition*. Journal of Physics: Conference Series, 1529(4). IOPscience

- Sarimole, F. M. & Septianto, A. E., 2024. *Implementation of IoT-Based Facial Recognition for Home Security System Using Raspberry Pi and Mobile Application*. International Journal Software Engineering and Computer Science (IJSECS), 4(2), pp.453–462. Lembaga KITA Journal

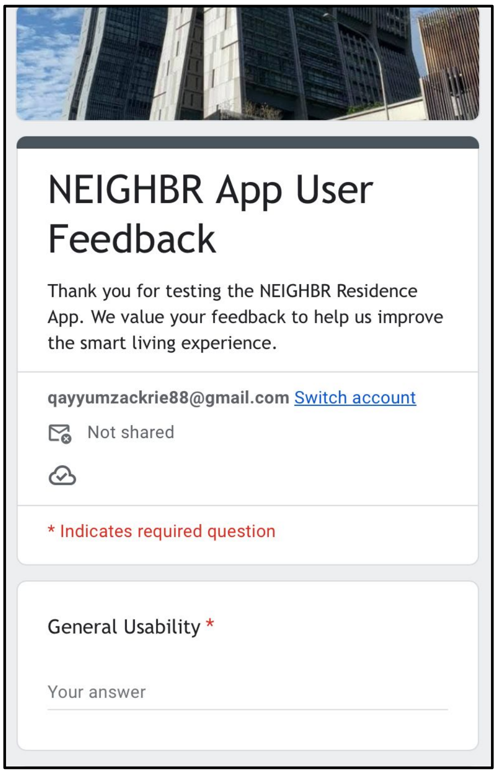
- TimeTec, 2021. *E-Park Penang Digitalizes Visitor Management with iNeighbour*. TimeTec News. news.timeteccloud.com

- TimeTec, 2022. *Visitors Analysis for Residential Properties in Malaysia*. TimeTec Cloud. timeteccloud.com

APPENDICES

APPENDIX A: Gantt Chart

APPENDIX B: User Acceptance Test Questionnaire



NEIGHBR App User Feedback

Thank you for testing the NEIGHBR Residence App. We value your feedback to help us improve the smart living experience.

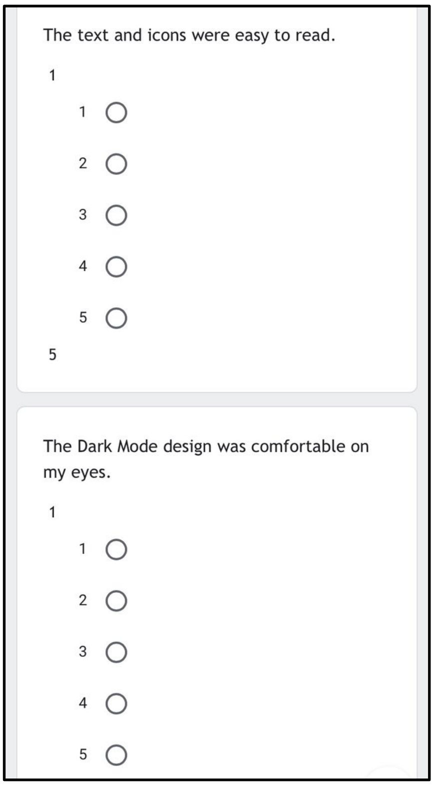
qayyumzackrie88@gmail.com Switch account

Not shared

* Indicates required question

General Usability *

Your answer



The text and icons were easy to read.

1

1

2

3

4

5

5

The Dark Mode design was comfortable on my eyes.

1

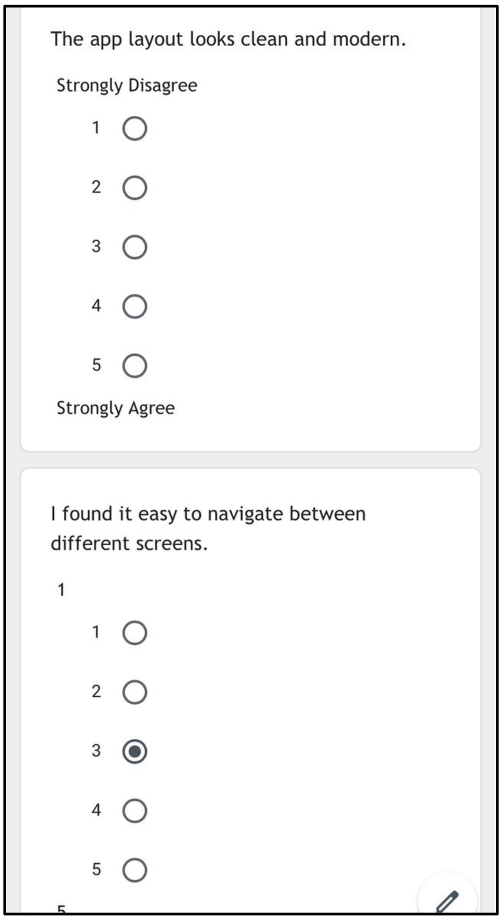
1

2

3

4

5



The app layout looks clean and modern.

Strongly Disagree

1 ○

2 ○

3 ○

4 ○

5 ○

Strongly Agree

I found it easy to navigate between different screens.

1

1 ○

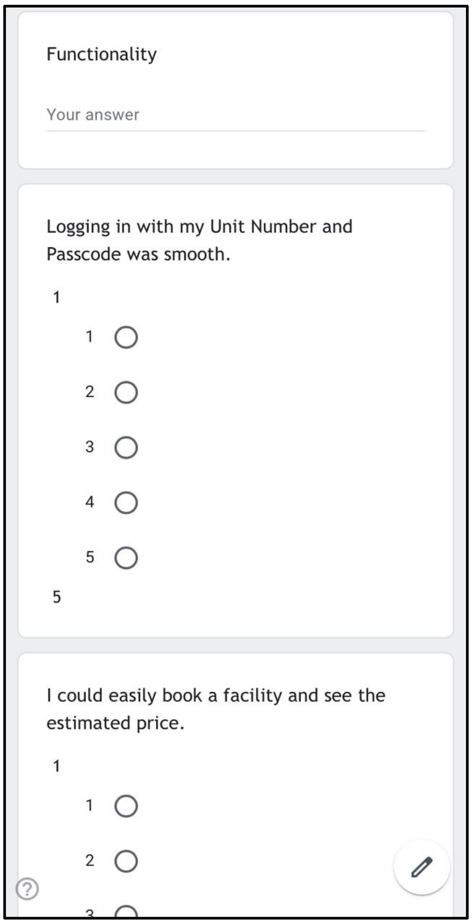
2 ○

3 ◉

4 ○

5 ○

5



Functionality

Your answer

Logging in with my Unit Number and Passcode was smooth.

1

1 ○

2 ○

3 ○

4 ○

5 ○

5

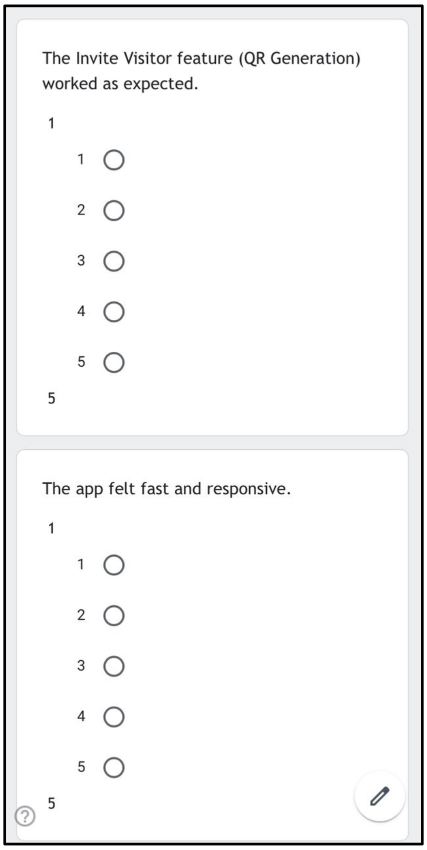
I could easily book a facility and see the estimated price.

1

1 ○

2 ○

3 ○



The Invite Visitor feature (QR Generation) worked as expected.

1

1
2
3
4
5

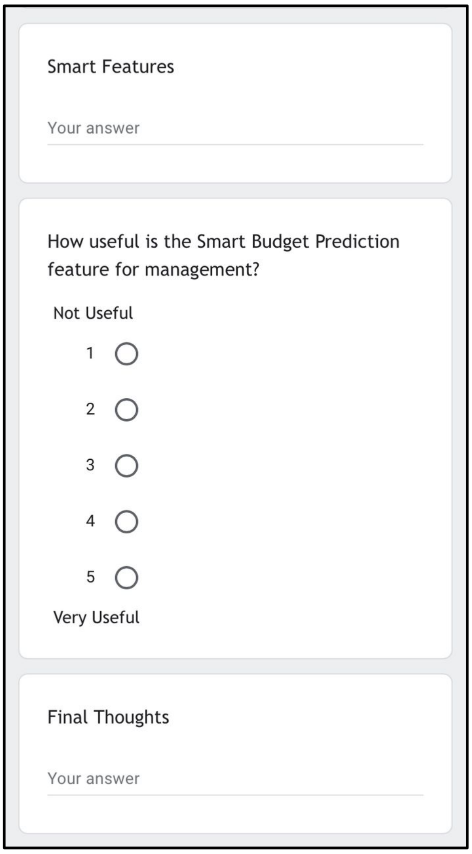
5

The app felt fast and responsive.

1

1
2
3
4
5

5



Smart Features

Your answer

How useful is the Smart Budget Prediction feature for management?

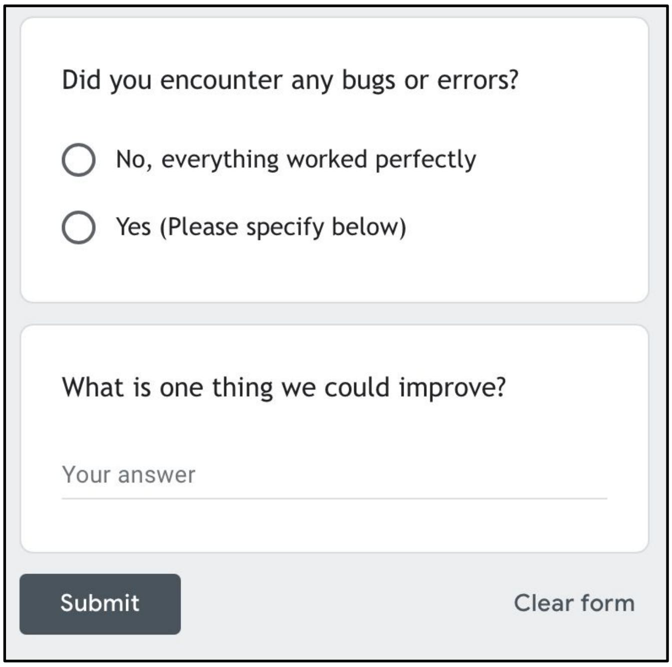
Not Useful

1
2
3
4
5

Very Useful

Final Thoughts

Your answer



Did you encounter any bugs or errors?

- No, everything worked perfectly
- Yes (Please specify below)

What is one thing we could improve?

Your answer

Submit

Clear form