

Sabil Al-Hikmah Claim Management System

Muhammad Daniel Naswan¹, Ruhaya Ab Aziz^{1*}

¹ *Fakulti Sains Komputer dan Teknologi Maklumat,*

Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: ruhaya@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2025.06.02.061>

Article Info

Received: 15 June 2025

Accepted: 20 November 2025

Available online: 30 November 2025

Keywords

Vendor Rating System, Digital Claim Processing, Student Meal Management System, Role-Based Access Control, RESTful API

Abstract

This project aims to modernize the process of claiming free meals at Universiti Tun Hussein Onn Malaysia (UTHM) by replacing the manual, paper-based system with a robust digital solution. Inefficiencies, inaccuracies, and delays in the existing system affect students, administrators, and vendors, necessitating a more streamlined approach. The proposed system integrates QR code technology, a mobile application, and a centralized web-based platform to facilitate efficient meal tracking, real-time data management, and transparent claim processing. Security features such as login authentication, data encryption, and role-based access controls ensure data integrity and user safety. Designed for UTHM's campus environment, the system targets eligible students, administrators, and approved vendors, enhancing operational workflows and reducing manual workload. Upon completion, the project will deliver a secure, efficient, and transparent solution that minimizes errors, reduces delays, and supports timely reimbursements, aligning with UTHM's commitment to student welfare and technological advancement.

1. Introduction

The Pay Sabil Al-Hikmah project represents a significant advancement in modernizing student welfare management at Universiti Tun Hussein Onn (UTHM). Historically, students claiming free meals on campus relied on manual processes involving paper-based attendance forms, which led to numerous challenges including human error, data duplication, and potential record loss. These inefficiencies created substantial administrative burdens and delayed crucial processes such as report generation and vendor reimbursement. To address these challenges, this project introduces a comprehensive digital solution that leverages QR code technology, mobile applications, and a centralized web-based platform. The system seamlessly enables students to claim meals efficiently while providing administrators with real-time monitoring capabilities.

1.1 Objective

This section presents the key goals of the Sabil Al-Hikmah system, which are to design an object-oriented solution, develop a mobile-based platform, and validate its functionality through user acceptance testing.

1. To design the Sabil al-Hikmah system based on object-oriented approach.
2. To develop a mobile-based Sabil al-Hikmah system.
3. To test the Sabil al-Hikmah system using user acceptance testing.

2. Literature Review

The *Sabil Al-Hikmah* system is designed to digitize and streamline the distribution of free student meals at UTHM by addressing key limitations of manual processes, such as data redundancy, lack of transparency, and administrative inefficiency. To fulfill this role, the system must incorporate technologies that support fast, secure, and accountable operations. QR code technology enables efficient and error-free meal claim verification, reducing human input and improving accuracy [1]. Centralized management architecture is essential for integrating claims, vendor operations, and reporting into a unified platform, thereby supporting real-time monitoring and data consistency [5]. Ensuring data confidentiality and access control is critical in a student-facing system, and this is achieved through a role-based access control framework that regulates user permissions according to defined responsibilities [1]. In promoting service accountability, the system incorporates a rating feature that allows students to provide feedback on vendor performance, which administrators can use for quality enforcement [5]. Furthermore, the system must safeguard sensitive student and transaction data; this is supported by Liu's multi-tier encryption model that ensures strong data protection across all layers of operation [3].

2.1 Case Study

This study focuses on the implementation of the Pay Sabil Al-Hikmah system at Universiti Tun Hussein Onn Malaysia (UTHM), which serves as a targeted digital solution for managing free student meal distribution. The existing manual method, which relies heavily on physical records and tick-based verification, has proven inefficient due to issues such as data duplication, lack of transparency, and limited tracking capabilities. These shortcomings hinder timely reporting, complicated auditing, and increase the administrative burden.

The Pay Sabil Al-Hikmah system is designed to address these challenges by introducing a centralized digital platform that automates the meal claiming process through QR code scanning, real-time validation, and secure transaction logging. Key features include role-based access control for user authorization, a vendor rating mechanism for service feedback, and a multi-layer encryption model to safeguard sensitive data. Through this case study, the system demonstrates how educational institutions can leverage modern technologies to enhance operational efficiency, data accuracy, and service accountability in student welfare programs.

2.2 Method and Technology

The Sabil Al-Hikmah system at UTHM integrates several key technologies to modernize and secure the meal claim process. The use of QR code technology has significantly enhanced transaction accuracy and efficiency by generating unique identifiers for each meal claim, enabling instant validation and eliminating manual entry errors [1]. This transition from paper-based to digital record-keeping ensures fast, seamless, and traceable processing for students and vendors. Central to the system is a centralized meal management platform, which consolidates claim processing, vendor oversight, and real-time reporting into a single data source, thereby reducing data inconsistencies and improving administrative decision-making [5]. To maintain secure and efficient operations, the system employs Role-Based Access Control (RBAC), which enforces strict access privileges based on user roles—students can submit claims and feedback, vendors manage their services, and administrators oversee all operations [1].

A built-in rating system promotes accountability by allowing students to evaluate vendors on food quality, service speed, and hygiene, offering both quantitative and qualitative feedback that is directly accessible to vendors for service improvement [5]. Vendor management is further supported through dedicated tools for service tracking, profile updates, and performance analysis, enabling administrators to monitor vendor compliance and intervene, when necessary, thereby ensuring consistent service quality [5]. Underpinning all of this is a robust data encryption framework that combines AES and RSA encryption to protect user accounts, sessions, and stored data, along with detailed audit logs for tracking and responding to security events efficiently [3]. Collectively, these integrated features create a secure, transparent, and efficient system that enhances the overall experience for students, vendors, and administrators.

2.3 Comparison with Existing System

Table 1 Comparison with Existing System

Feature	Sabil Al-Hikmah	MySchoolBucks	CharityTracker	Boost e-Wallet
QR-Code Technology	✓	✓	-	✓
Centralized Meal Management	✓	✓	-	-
Role-Based Access	✓	✓	✓	✓

Control				
Rating System	✓	-	-	✓
Data Encryption and Security	✓	✓	✓	✓

Table 1 compares key features across four systems, demonstrating that Sabil Al-Hikmah offers comprehensive functionality including QR technology, meal management, access control, ratings, and security measures, like existing solutions.

3. Methodology

The project follows a structured waterfall methodology that ensures systematic development progression. As outlined by Saravanos & Curinga, the planning phase establishes project scope and risk assessment frameworks, leading into a comprehensive analysis phase where both functional and non-functional requirements are documented [6]. Murthy emphasizes the importance of this transition into the design phase, where requirements are transformed into technical specifications [7]. The implementation and testing phases complete the development cycle, with careful attention to code quality and thorough system validation [6]

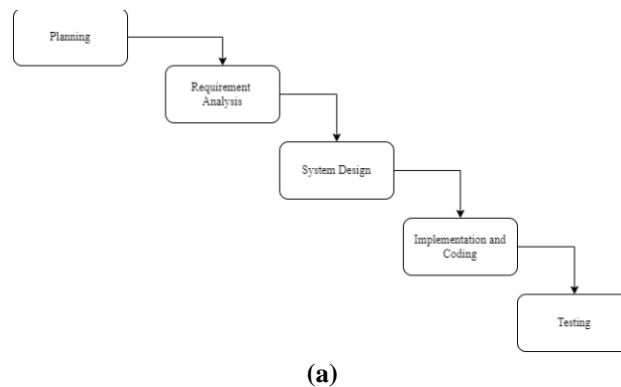


Fig 1. Waterfall Methodology [9]

Table 2 Software Process Model

Phase	Key Activities	Key Output	References
Planning	Stakeholder meetings to define objectives and deliverables Gantt chart for timeline planning (see Appendix A) Risk assessment and mitigation strategies	Clear project scope Development timeline Risk mitigation plan	[6] Saravanos & Curinga
Requirement Analysis	Requirement gathering and validation Stakeholder interviews Use-case development	Software Requirements Specification (SRS) Functional and non-functional requirement list	[7] Murthy
Design	System architecture design Database schema creation UI prototypes and technical diagrams	Technical specifications UI mockups Scalable and user-focused system design	[6] Saravanos & Curinga
Implementation	Module development Security and database integration Code reviews and documentation	Functional software modules Secure, documented, and integrated system	[7] Murthy
Testing	Unit and integration testing User Acceptance Testing (UAT)	Validated system functionality	[6] Saravanos & Curinga

Bug tracking and resolution	Stakeholder-approved system
	Test reports and bug fixes

4. Analysis and Design

The analysis and design phase converts theoretical concepts into practical system architecture. Functional requirements cover core features like user authentication, QR processing, and claim management, while non-functional requirements address security, accessibility, and performance. System analysis employs use case, activity, and sequence diagrams to map user interactions and data flows. The interface design creates an intuitive experience that meets stakeholder needs while maintaining security and efficiency.

4.1 Requirement Analysis

The requirement analysis for the Sabil Al-Hikmah system encompasses three main categories: functional, non-functional, and user requirements. The functional requirements include seven core modules: user authorization, QR scanning, student management, vendor management, claim management, rating system, and report generation. Non-functional requirements cover operational efficiency, security, usability, integrity, availability, and maintainability. User requirements detail specific needs for students, vendors, and administrators, ensuring comprehensive system functionality.

4.1.1 Functional Requirements

Table 3 Functional Requirements

Module	Requirement
Authorize User	The user shall be able to login and logout from the system.
	The system shall implement role-based access control for different user types.
	The system shall provide password management including reset and update capabilities.
	The system shall lock accounts after multiple failed login attempts.
	The system shall implement two-factor authentication when required.
	The system shall maintain audit logs of all security-related actions.
Make Sabil Transaction	The student shall be able to access the QR scanner feature through the system.
	The system shall be able to activate a device camera for scanning.
	The student shall be able to scan vendor-provided QR codes.
	The system shall validate QR code authenticity.
	The system shall display meal details after a successful scan.
	The system shall maintain a history of QR code scans.
Manage Student Information	The system shall provide a manual entry option when scanning is not possible.
	The administrator shall be able to add new student information to the system.
	The administrator shall be able to update existing student details.
	The administrator shall be able to view complete student profiles.
Manage Vendor Information	The administrator shall be able to delete student records when necessary.
	The student shall be able to submit a new application to the system.
	The admin shall be able to approve or reject submitted applications.
	The system shall maintain a complete history of all applications.
	The system shall automatically notify students of application status changes.

Manage Sabil Application	The administrator shall be able to add new vendor details to the system.
	The administrator shall be able to update existing vendor information.
	The administrator shall be able to view complete vendor profiles.
	The administrator shall be able to delete vendor records when necessary.
	The administrator shall be able to assign specific roles and permissions to vendors.
Rate Vendor	The student shall be able to submit ratings for vendor services.
	The system shall calculate and display average ratings for vendors.
	The students shall be able to give detailed feedback and comments.
	The system shall analyze and display trends in student satisfaction.
	The system shall show current ratings on vendor profiles.
Generate Reports	The administrator shall be able to generate financial reports.
	The administrator shall be able to create student claim summaries.
	The administrator shall be able to generate vendor performance reports.
	The system shall support exports to multiple formats, including PDF and Excel.
	The system shall allow filtering of reports by date range and other criteria.
	The system shall support scheduled automatic report generation and distribution.

4.1.2 Non-Functional Requirements

Table 4 Non-Functional Requirements

Module	Requirement
Operational	The system can be accessed and used in any modern web browser including Google Chrome, Mozilla Firefox, or Microsoft Edge.
Security	The system can only be accessed using authorized credentials including email and password. All data transmissions must be encrypted using industry-standard protocols.
Security	All types of users can easily understand and navigate the system interface with minimal training required.
Integrity	The system's database and all stored information will be properly maintained, backed up regularly, and protected from corruption or unauthorized modifications.
Availability	The system is designed to be operational and accessible 24/7 with minimal planned downtime for maintenance.
Maintainability	Regular system updates and maintenance will be performed at scheduled intervals to ensure optimal performance and reliability.

4.2 System Analysis

The system analysis for Sabil Al-Hikmah employs Unified Modeling Language (UML) diagrams to document system behavior and structure [10]. The analysis includes use case diagrams showing interactions between students, admins, and vendors with seven core functions. It features detailed use case specifications, activity diagrams, sequence diagrams, and a comprehensive class diagram. The database design encompasses 11 interconnected tables managing users, claims, ratings, and system operations.

4.2.1 Use Case Diagram

The use case diagram illustrates system interactions between students, vendors, and administrators, showing key functions like meal claims, ratings, and management processes. It highlights relationships between components for clear stakeholder understanding.

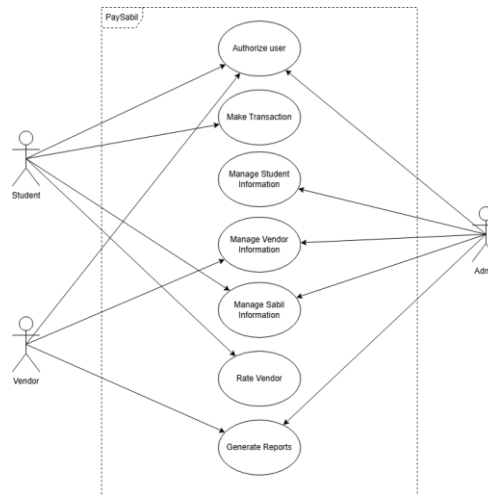


Fig. 2 Use Case Diagram for The Proposed system

4.2.2 Activity Diagram

Activity diagrams illustrate operational flows for key processes like claims, ratings, and reports, showing decision points and alternate paths to guide system development and maintenance.



Fig. 3 Activity Diagram for Make Sabil Transaction

Fig. 3 illustrates the *Make Transaction* activity diagram, showing the workflow from initiation through validation to completion, with decision points and parallel processes for transaction handling.

4.2.3 Sequence Diagram

The sequence diagram represents the dynamic behavior of the system by illustrating the interactions between objects over time. It shows how users and system components exchange messages to complete a process. In the Sabil Al-Hikmah system, this diagram is especially useful for visualizing how a student initiates a QR code scan, how the system validates the claim, and how the transaction is processed and stored.

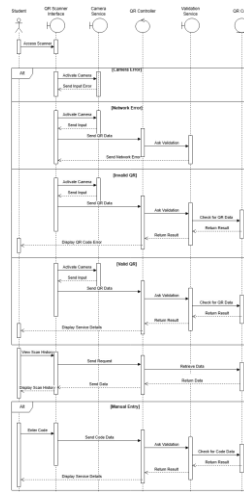


Fig. 4 Sequence Diagram for *Make Sabil Transaction*

Fig. 4 illustrates the interactions and message flow between system components during transaction processing, showing the temporal order of operations and system responses.

4.3 General System Architecture

The general system architecture presents an overview of the system’s structural design. It outlines how various components are organized and interact, including the user interfaces, application logic, and database layer. In this system, a layered architecture is used where students, vendors, and administrators are all derived from a base User class. Each of these user types interacts with a centralized backend, enabling secure, role-based operations while maintaining modularity and scalability.

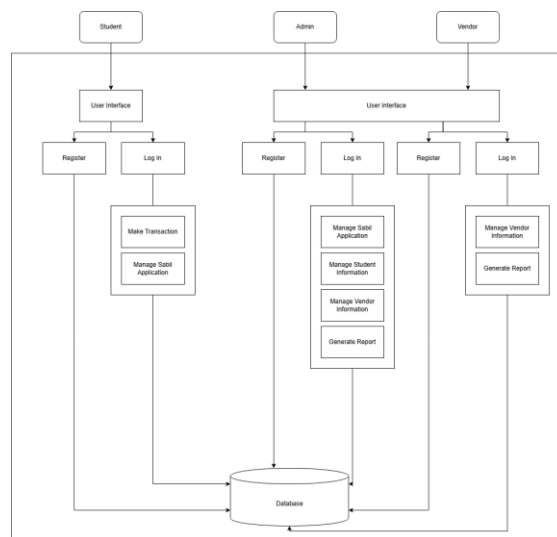


Fig. 5 General System Architecture

Fig. 5 diagram illustrates the hierarchical structure and relationships between system components, showing the inheritance from a base User class to specialized Student, Admin, and Vendor classes. Each class contains relevant attributes and methods, with connections to the central database for data management.

4.4 Class Diagram

A class diagram is a static structure diagram that describes the classes in a system, their attributes, operations, and the relationships among them. For the Sabil Al-Hikmah system, the class diagram defines the core data models such as User, Student, Vendor, Admin, Transaction, Rating, and others. This diagram serves as the blueprint for both the database schema and application logic, ensuring consistency and integrity throughout development process.

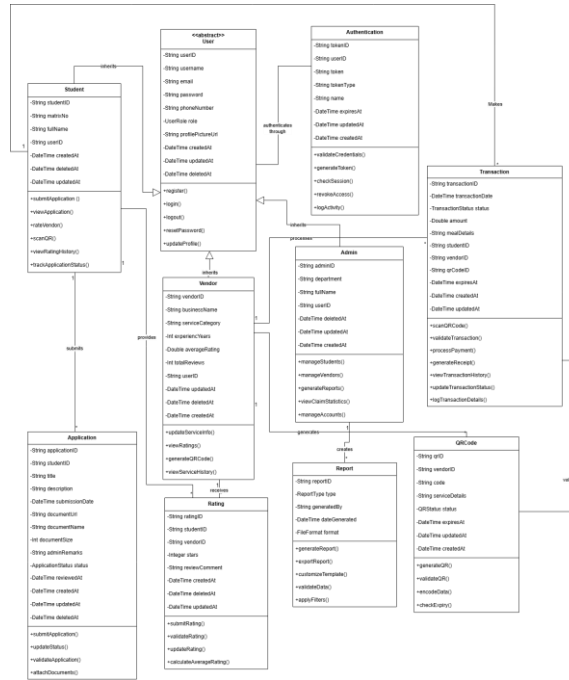


Fig. 6 Class Diagram of the Proposed System

Figure 6 shows the Class Diagram of the Sabili Al-Hikmah Claim Management System, which consists of normalized and interconnected tables representing users, students, vendors, admins, transactions, and QR codes. The User table serves as the core entity linking to role-specific tables such as Student, Vendor, and Admin. Other key tables include Application for claim requests, Transaction for QR-based meal claims, Rating for vendor feedback, Report for performance and financial data, Authentication for session control, Service for vendor offerings, QRCode for meal validation, and Documents for file uploads. This structure supports secure, efficient, and role-based operations across all modules.

4.5 Database Design

The database design lays out the structure and relationships of the system's data. It defines how information is stored, accessed, and managed across various modules. The Sabili Al-Hikmah system uses a normalized relational database to store user data, transaction records, vendor details, application statuses, and more. This approach ensures data integrity, reduces redundancy, and supports efficient query execution.

Table 5 Database Schema of the Proposed System

Collection	Attributes
User	userId, username, email, password, phone_number, role, created_at, updated_at
Student	studentID, full_name, matrix_no, profile_picture_url, userID
Vendor	vendorID, business_name, service_category, experience_years, average_rating, total_reviews, profile_picture_url, userID
Admin	adminID, full_name, department, profile_picture_url, userID
Application	applicationID, status, submission_date, document_url, created_at, updated_at, studentID
Rating	ratingID, stars, review_comment, vendor_response, review_date, response_date, is_modified, claimID, studentID, vendorID
Report	reportID, type, generated_date, format, parameters, adminID
Authentication	sessionID, token_expiry, last_activity, userID
Documents	documentID, filename, file_type, file_size, uploaded_at, claimID
Service	serviceID, name, description, price, category, preparation_time, is_available, created_at, updated_at, vendorID

Transaction	transactionID, studentID, vendorID, qrCodeID, transactionDate, status, amount, mealDetails, created_at, updated_at
QRCode	qrCodeId, service_details, generated_date, expiry_date, status, vendorId, serviceId

4.6 Design Interface

The user interface design for Sabil Al-Hikmah demonstrates the visual layout and interaction flow of the system. It showcases key interfaces including login pages for students and administrators, registration forms, main dashboards, claim management screens, and QR code generation functionality. The design focuses on creating an intuitive, user-friendly experience while maintaining consistency across different user roles.

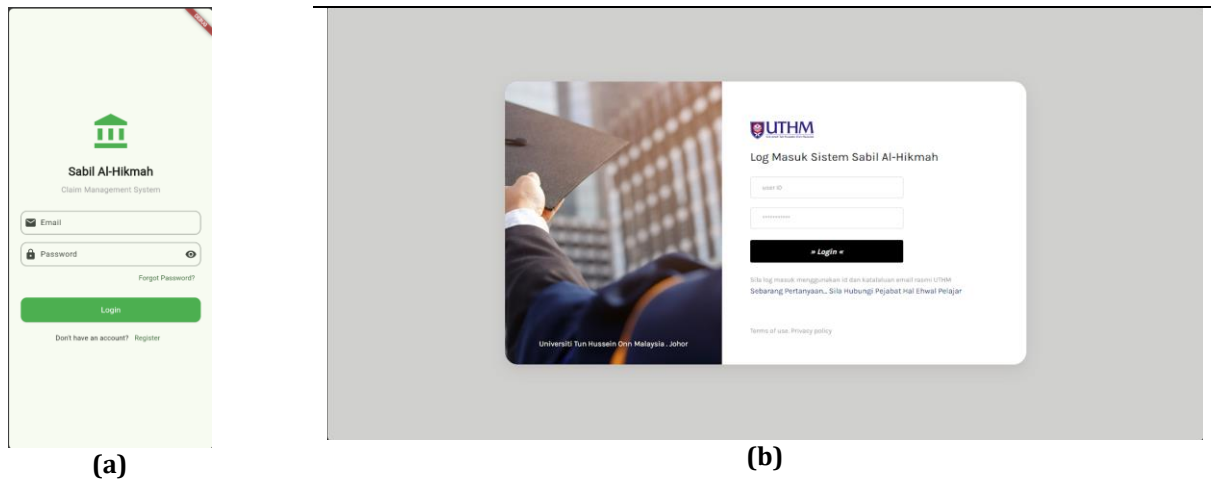


Fig. 7 (a) Login Page (Student) and (b) Login Page (Admin & Vendor)

Fig. 7 presents two distinct login interfaces for the Sabil Al-Hikmah system: (a) displays a minimalist student login page with green accent colors and basic input fields, while (b) shows the administrator and vendor login interface incorporating UTHM's institutional branding and design elements.

5. Implementation and Testing

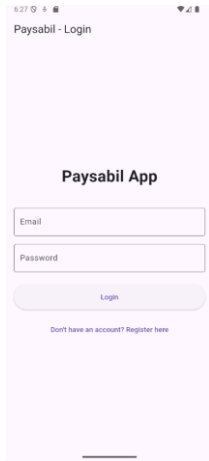
This chapter describes how the Sabil Al-Hikmah system was built and validated. It covers the implementation of each system module, presenting user interface designs and back-end integrations. It also discusses the testing strategies used to ensure the system's functionality, security, and usability. The goal is to demonstrate that the system operates as intended and meets user requirements.

5.1 Implementation

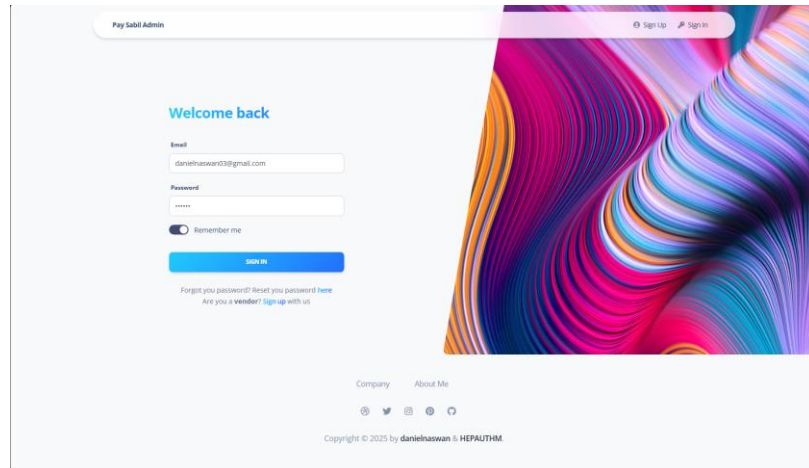
This section describes the development of functional modules in the PaySabil system, detailing how each feature was implemented to create a comprehensive and user-friendly experience.

5.1.1 Authenticate User Interface

Authentication is a critical module that ensures only authorized users can access the system. The interface includes login and registration forms tailored to students, vendors, and administrators. These forms validate user credentials, implement role-based access control, and manage secure session handling.



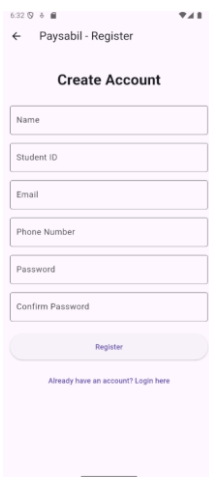
(a)



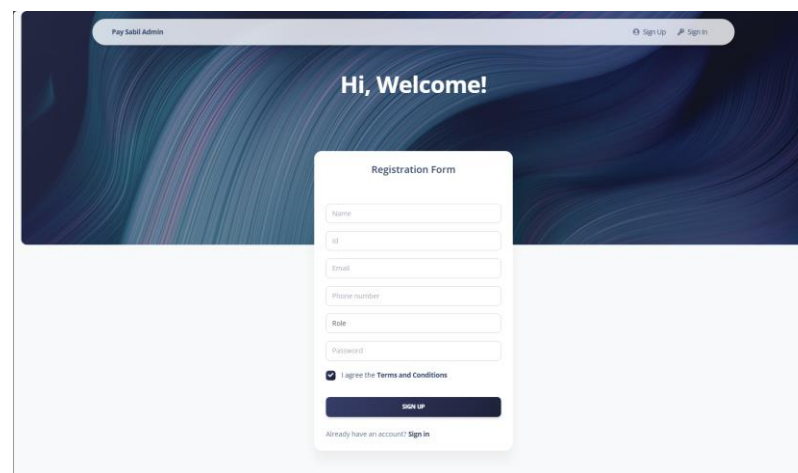
(b)

Fig. 8 (a) Login Page (Student) and **(b)** Login Page (Admin & Vendor)

Figure 8 (a) shows the login page interface for students. It provides a clean interface for students to enter their credentials before accessing the system. Figure 8 (b) shows the login interface for both admin and vendor users. This version integrates role-specific redirection logic after authentication to ensure correct dashboard access.



(a)



(b)

Fig. 9 (a) Register Page (Student) and **(b)** Register Page (Vendor)

Figure 9 (a) shows the registration page for students. The system checks the student's email for duplicates before storing the data into the MySQL database by sending JSON packet through HTTP request and send it to backend. Figure 9 (b) shows the vendor registration page. After entering required business details, vendor applications are stored in a pending state for admin approval.

5.1.2 Manage Student Information Interface

This module allows administrators to manage student data efficiently. The interface includes features for creating, reading, updating, and deleting student profiles. It ensures accurate record-keeping and supports seamless integration with the claim tracking system.

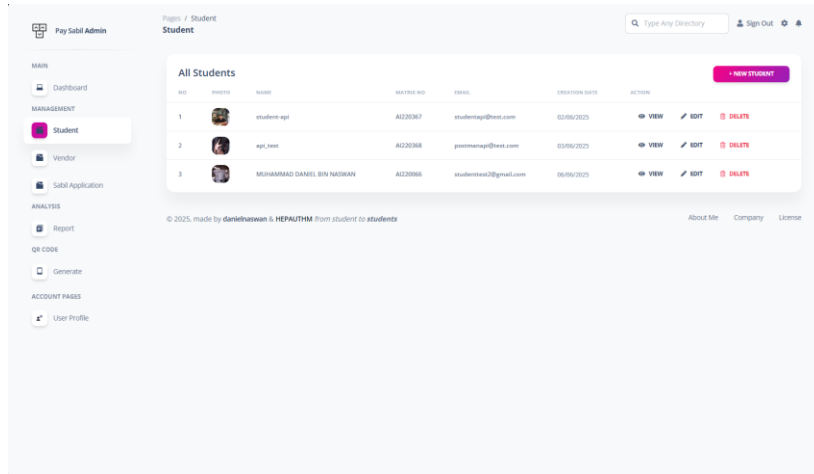


Fig. 10 Manage Student Page

Figure 10 shows the admin interface to manage student records. Admins can perform CRUD operations such as adding, editing, viewing, and deleting student profiles. Each student record includes personal information, matric number, and contact details.

5.1.3 Manage Vendor Information Interface

Through this interface, administrators can manage vendor profiles, including updating business details and assigning access roles. It provides transparency and control over the vendor ecosystem involved in the meal distribution program.

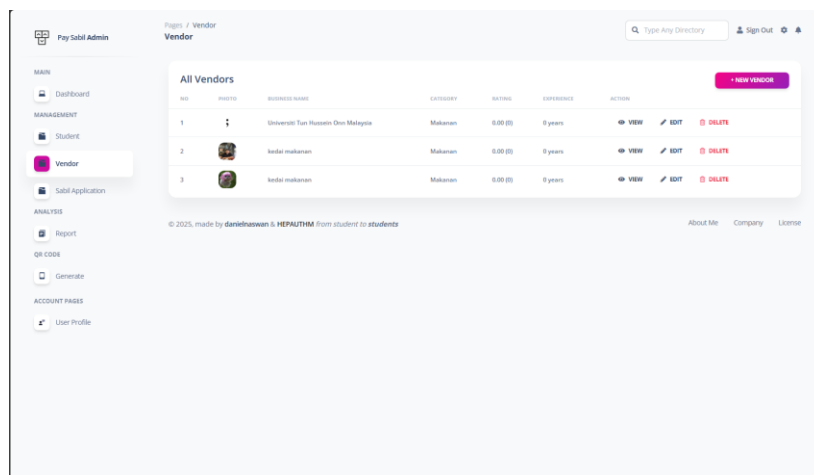
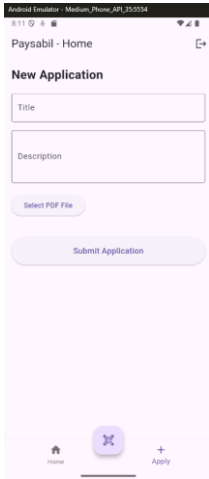


Fig. 11 Manage Vendor Page

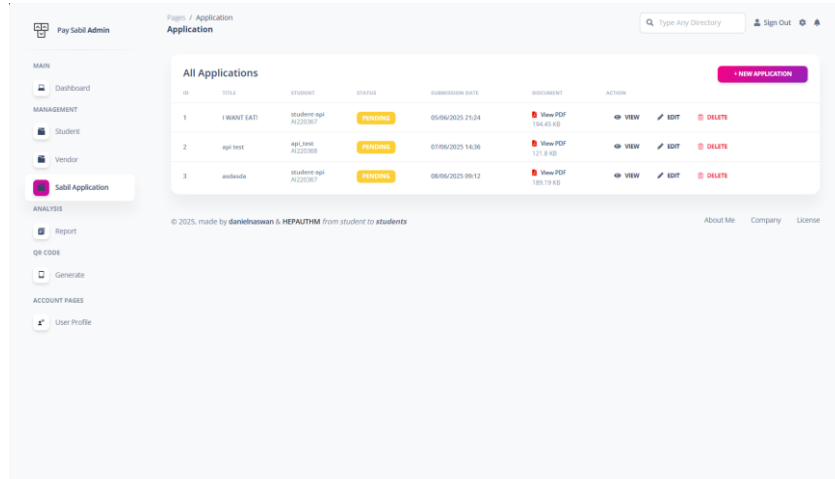
Figure 11 displays the vendor management interface, where the admin can view all registered vendors. Admins have the ability to update vendor profiles, assign roles, or remove vendor access from the system.

5.1.4 Manage Sabil Application Interface

This interface supports both student and admin roles. Students can submit applications for the Sabil program and upload supporting documents, while admins review, approve, or reject these applications. The module ensures a structured, trackable workflow for application processing.



(a)



(b)

Fig. 12 (a) Sabil Application Screen (Student) and **(b)** Sabil Application Page (Admin)

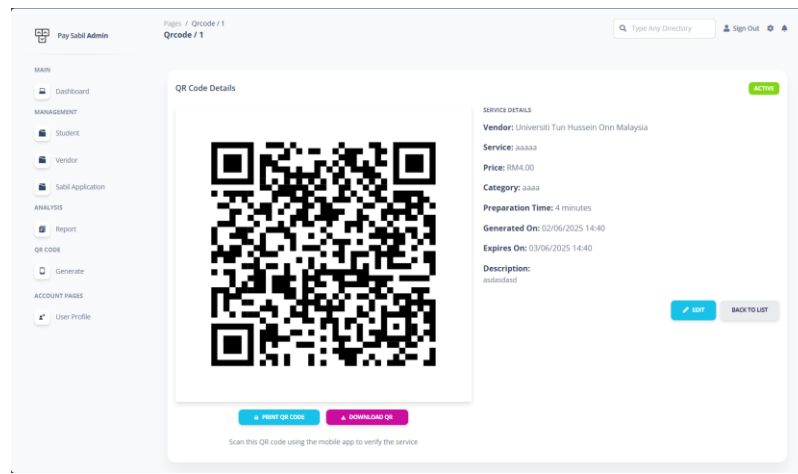
Figure 12 (a) shows the student interface for submitting a new application to join the PaySabil program. Students can upload documents and monitor the application status. Figure 12 (b) shows the admin view of all submitted applications. Admins can approve or reject applications. The system automatically updates the student via notification and maintains a history of all decisions.

5.1.5 Make Sabil Transaction Interface

This interface facilitates real-time meal transactions using QR codes. Students scan codes generated by vendors, which are then validated by the system. The interface logs the transaction, displays meal details, and ensures secure and traceable processing.



(a)



(b)

Fig. 13 (a) QR Scanner Page (Student) and **(b)** QR Generator Page (Admin)

Figure 13 (a) displays the QR scanner interface on the student side. When a QR code is scanned, the system checks for code authenticity, retrieves meal details, and records the transaction in the database. Figure 13 (b) shows the QR code generator for vendors or admins. Each QR code includes embedded data for vendor ID, service ID, and expiration timestamp. This allows real-time verification during scanning.

5.1.6 Rate Vendor Interface

Vendor rating enhances service accountability. This module allows students to rate vendors based on service quality, hygiene, and meal satisfaction. The system calculates average ratings and provides trend analysis to administrators for quality control.



Fig. 14 (a) Sidebar Interface, (b) List of Vendor and (c) Rating Page

Figure 14 (a) shows the sidebar interface with navigation to vendor rating features. Figure 14 (b) presents the list of vendors eligible for rating. Students can select a vendor to provide feedback. Figure 14 (c) displays the vendor rating form. Students can submit a star rating and detailed comments. The system calculates the average rating and displays it on vendor profiles for transparency.

5.1.7 Generate Reports Interface

The reporting interface gives administrators the tools to generate detailed insights on system usage. It includes financial reports, student participation logs, anomaly detection, and feedback summaries. These reports support strategic decision-making and operational oversight.

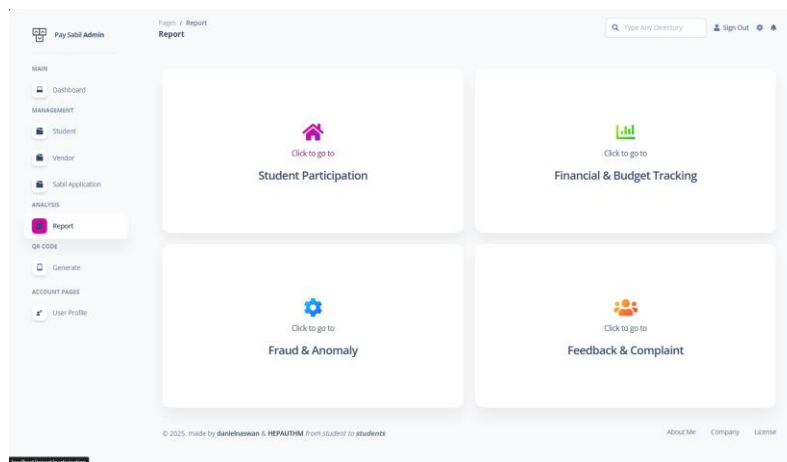
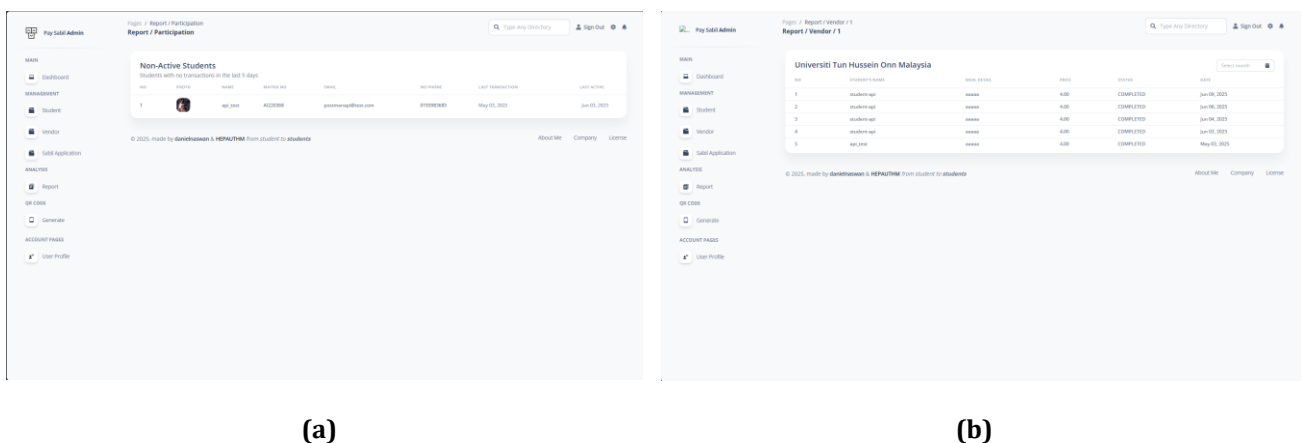
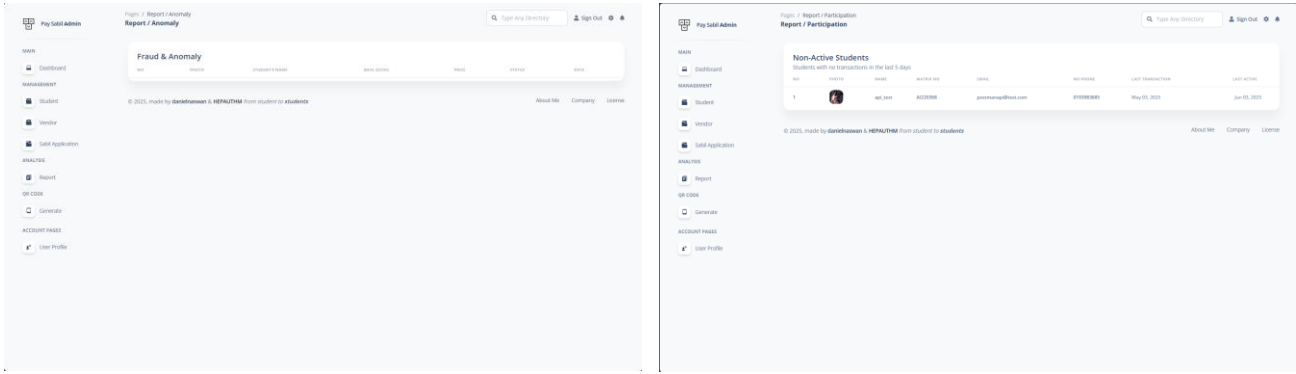


Fig. 15 Report Selection Menu



(a)

(b)



(c)

(d)

Fig. 16 (a) Student Participation Page, (b) Financial Report Page, (c) Anomaly Report Page and (d) Student Feedback Page

Figure 15 presents the main reporting interface where admins can choose from various report types, such as financial summaries and participation data. Figure 16 (a) shows the Student Participation Report; detailing meal claim activity for each student. Figure 16 (b) displays the Financial Report, summarizing transactions by date, vendor, and claim value. Figure 16 (c) shows the Anomaly Report, which flags any irregular transactions such as duplicate claims. Figure 16 (d) presents the Student Feedback Report, summarizing student reviews and ratings for vendors. This aids in vendor performance evaluation and service improvement.

5.2 Testing Result

This section presents the outcomes of functional testing conducted on each module of the system. It outlines test cases, expected results, and actual outcomes. Functional testing verifies that all features operate as intended under normal usage scenarios. Each module passed its respective tests, confirming system readiness and stability.

Table 6

Module	Description	Expected Result	Actual Result	Pass/Fail
Authorize User	To check login/logout functionality	User can login and logout successfully	User successfully logs in and logs out	Pass
	To check role-based access control	User access is limited based on role	System correctly restricts access based on user roles	Pass
	To check password management features	User can reset and update password	User successfully resets and updates password	Pass
	To check audit logging	All security-related actions are logged	Audit logs are generated for login, logout, reset, and updates	Pass
Make Sabil Transaction	To check QR scanner accessibility	Student can access QR scanner feature	QR scanner accessed from system interface	Pass
	To check camera activation	Device camera activates for QR scanning	Camera activates successfully	Pass
	To check QR code scanning and validation	Student can scan and system validates QR code	QR code scanned and validated successfully	Pass
	To check meal detail display	System displays meal details after scan	Meal details shown after QR validation	Pass
	To check scan history feature	System maintains history of scanned QR codes	Scanned QR history is displayed correctly	Pass
	To check manual QR entry	Manual entry option is available when scan fails	Manual entry works as expected	Pass
Manage Student Information	To check student record management by admin	Admin can add, update, view, and delete student records	Admin performs all CRUD operations on student records successfully	Pass

Manage Vendor Information	To check student application submission	Student can submit application	Application submitted successfully	Pass
	To check admin approval/rejection	Admin can approve or reject applications	Admin manages application statuses	Pass
	To check application history	System maintains complete application history	History of applications displayed correctly	Pass
Manage Sabil Application	To check vendor record management by admin	Admin can add, update, view, delete vendor records	Admin manages vendor information	Pass
	To check role assignment	Admin assigns roles and permissions to vendors	Vendor roles and permissions assigned successfully	Pass
Rate Vendor	To check rating and feedback submission by student	Student can rate and give feedback on vendor services	Rating and feedback submitted successfully	Pass
	To check average rating calculation	System calculates and displays average ratings	Average rating displayed correctly	Pass
	To check trend analysis	System shows trends in student satisfaction	Trend analytics displayed as expected	Pass
Generate Reports	To check financial, claim, and vendor reports generation	Admin can generate all required reports	All reports generated as expected	Pass
	To check export formats	Reports exportable in PDF and Excel	Reports exported successfully	Pass
	To check filtering and scheduling	Reports can be filtered and scheduled	Filter and scheduling features work correctly	Pass

5.3 User Acceptance Testing

User Acceptance Testing (UAT) measures the system’s performance from the end-user perspective. It involves real users interacting with the system to evaluate its usability, efficiency, and reliability. The UAT results provide feedback on areas such as login experience, feature completeness, ease of navigation, and overall satisfaction. This testing ensures the system meets user expectations before full deployment.

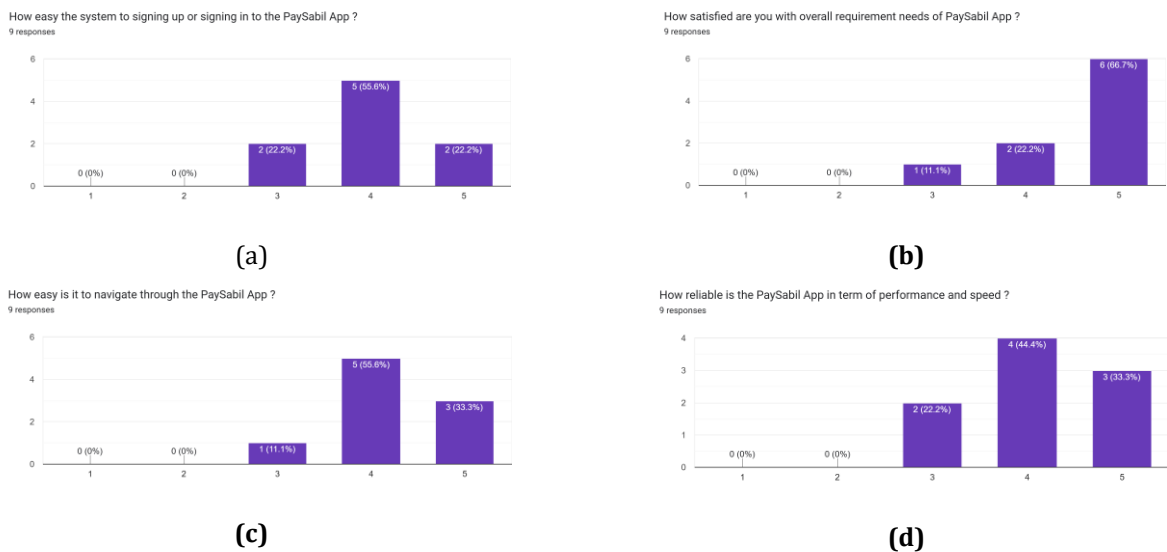


Fig. 17 (a) (b) (c) (d)

The sign-up and sign-in process received strong approval based on Figure 17 (a), with 55.6% of users rating it as 5 (strongly agree) and 22.2% as 4 (agree). Only 22.2% rated it as 3 (neutral), and no respondents selected lower scores. This indicates that the authentication system is user-friendly and efficient, contributing positively to the overall user experience. The absence of negative ratings reinforces that this aspect of the app is well-designed and meets user expectations. Based on Figure 17 (b) A significant majority (66.7%) rated their satisfaction with the app's ability to meet their needs as 4 (agree), demonstrating strong approval. However, 22.2% chose 2 (disagree) and 11.1% selected 1 (strongly disagree), suggesting that while most users find the app effective, a minority feel it does not fully address their requirements. This divergence may indicate the need for additional features or customization options to cater to a broader range of user needs. The survey results in Figure 17 (c) indicate that users generally find the PaySabil App moderately easy to navigate. A majority (55.6%) rated navigation as 3 (neutral), while 33.3% rated it as 4 (agree), suggesting that most users experience satisfactory navigation. However, 11.1% selected 2 (disagree), indicating that a small portion of users encountered difficulties. This suggests that while the app's navigation is functional, there may be room for improvement in terms of intuitiveness and user flow to enhance the experience for all users. Performance and speed received moderate ratings based on Figure 17 (d), with 44.4% selecting 4 (agree) and 33.3% choosing 3 (neutral). Another 22.2% rated it as 2 (disagree), but no respondents selected extreme scores (1 or 5). This suggests that while the app performs adequately for most users, there is a perception that it could be more responsive or stable. Optimizing backend processes and reducing latency could help improve these ratings.

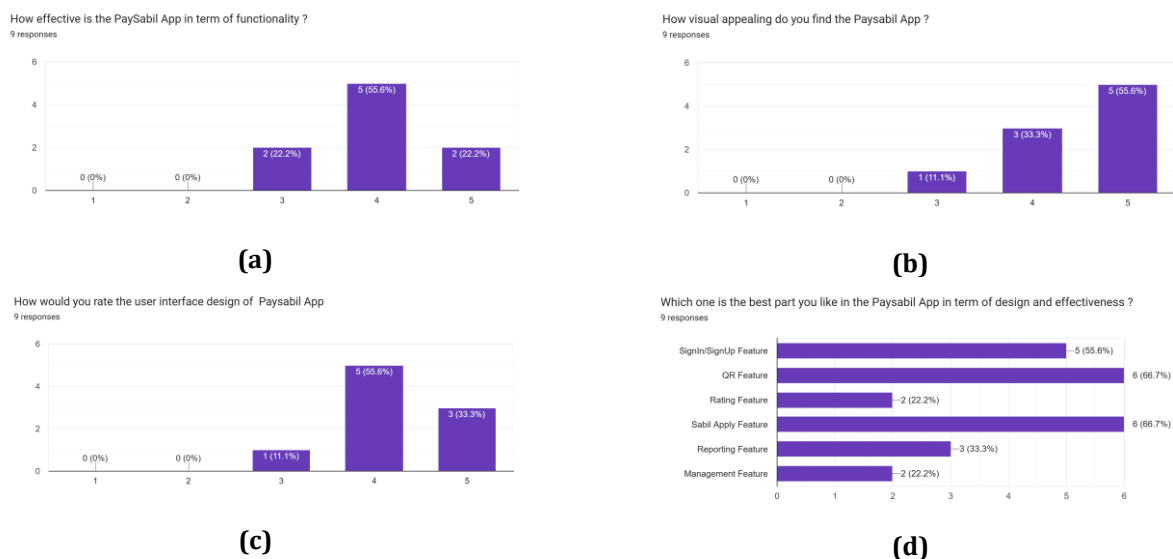


Fig. 18 (a) (b) (c) (d)

Based on Figure 18 (a) Functionality was rated as 3 (neutral) by 55.6% of users, while 22.2% each selected 2 (disagree) and 4 (agree). The absence of extreme scores (1 or 5) suggests that while the app's features work as intended, they may not exceed user expectations. Introducing more advanced or seamless functionalities could help elevate user satisfaction in this area. Responses regarding visual appeal were polarizing based on Figure 18 (b), with 35.6% selecting 5 (strongly agree) and 33.3% choosing 3 (neutral). However, 11.1% rated it as 1 (strongly disagree), indicating a notable dissatisfaction among a subset of users. This suggests that while many users appreciate the app's visual style, others may find it lacking in modern design elements or consistency. Enhancing color schemes, typography, and graphic elements could help bridge this gap. The app's interface design received mixed feedback as we can see in Figure 18 (c), with 35.6% rating it as 5 (strongly agree) and 33.3% as 3 (neutral). The absence of ratings at the lower end (1 or 2) suggests that while some users find the design highly appealing, others remain indifferent. This indicates that the UI is generally well-received but may benefit from refinements to ensure broader aesthetic appeal and usability. When asked about their favorite features, 66.7% selected the "QR Feature" as the best, followed by 55.6% favoring the "Sign In/Sign Up Feature." Other features, such as "Rating," "Sabil Apply," and "Reporting," received lower but notable approval as shown in Figure 18 (d). This indicates that users highly value core functionalities, while secondary features may require further refinement to enhance their perceived usefulness.

6. Conclusion

In conclusion, the Sabil Al-Hikmah Claim Management System successfully addresses the inefficiencies of the traditional paper-based meal claim process at Universiti Tun Hussein Onn Malaysia (UTHM) by introducing a

secure, efficient, and transparent digital solution. Leveraging QR code technology, a mobile application, and a centralized web-based platform, the system streamlines meal tracking, real-time data management, and claim processing for students, administrators, and vendors. Key features such as role-based access control, data encryption, and a vendor rating system ensure security, accountability, and user satisfaction.

The system’s implementation aligns with UTHM’s commitment to technological advancement and student welfare, significantly reducing manual errors, administrative burdens, and processing delays. User acceptance testing demonstrated positive feedback on core functionalities like QR code scanning, authentication, and reporting, though opportunities for improvement in navigation and performance were identified.

For future enhancements, the system could integrate advanced analytics for predictive insights, expand notification features for real-time updates, and further optimize the user interface for broader accessibility. Additionally, exploring integration with other campus systems could enhance interoperability and scalability. Overall, the Sabil Al-Hikmah system represents a transformative step toward modernizing student welfare management, fostering efficiency, transparency, and trust within the UTHM community.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The authors confirm contribution to the paper as follows: **Study Conception and Design:** Muhammad Daniel Naswan, Ruhaya Binti Ab Aziz; **Data Collection:** Muhammad Daniel Bin Naswan, Ruhaya Binti Ab Aziz; **Analysis and Interpretation of Results:** Muhammad Daniel Bin Naswan, Ruhaya Binti Ab Aziz; **Draft Manuscript Preparation:** Muhammad Daniel Bin Naswan, Ruhaya Binti Ab Aziz. All authors reviewed the results and approved the final version of the manuscript.

Appendix A: Project Timeline

PROJECT TITLE: MUHAMMAD DANIEL BIN NASWAN					COMPANY NAME: FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT												
PROJECT MANAGER: RUHAYA BINTI AB AZIZ					DATE: 10/6/24												
TASK TITLE	START DATE	DUE DATE	DURATION (week)	PCT OF TASK COMPLETE	October				November				December				January
					WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10	WEEK 11	WEEK 12	WEEK 13
Planning Phase																	
Define Project Scope and Objectives	10/02/24	10/12/24	1	100%	█	█	█	█									
Prepare Proposal	10/02/24	10/12/24	1	100%	█	█	█	█									
Complete Chapter 1	10/02/24	10/12/24	1	100%	█	█	█	█									
Research Phase																	
Review Research Journals and Past Work	10/12/24	10/12/24	1	100%													
Develop Chapter 2 (Literature Review)	10/12/24	11/12/24	1	100%													
Design Phase																	
Design System Architecture	11/12/24	11/12/24	1	100%													
Develop Chapter 3 (Methodology)	11/12/24	11/12/24	1	100%													
Develop Chapter 4 (System Analysis and Design)	11/12/24	12/12/24	1	100%													
Development Phase																	
Preparing Module for prototype	12/12/24	12/12/24	1	100%													
Develop Prototype	12/12/24	12/12/24	1	100%													
Documentation Phase																	
Complete Chapters 5-8	12/12/24	12/12/24	1	100%													
Submit Final FYP 1 Report	12/12/24	12/12/24	1	100%													

References

Journal

- [1] Wong, C. C., Chong, L. Y., Chong, S. C., & Law, C. Y. (2023). QR Food Ordering System with Data Analytics. *Journal of Informatics and Web Engineering*, 2(2), 249-272.
- [2] Ferraiolo, D., Cugini, J., & Kuhn, D. R. (1995, December). Role-based access control (RBAC): Features and motivations. In *Proceedings of 11th annual computer security application conference* (pp. 241-48).
- [3] Liu, G. (2022). The application of data encryption technology in computer network communication security. *Mobile Information Systems*, 2022(1), 3632298.

- [4] Bhandge, K., Shinde, T., Ingale, D., Solanki, N., & Totare, R. (2023). A proposed system for touchpad electronic ordering. *International Journal of Scientific Research in Science, Engineering and Technology*, 10(2), 157-163.
- [5] Nikose, A., et al. (2023). Design and implementation of a digital dining system using QR code technology. *International Journal of Scientific Research in Science, Engineering and Technology*, 10(2), 157-163.
- [6] Saravanos, A., & Curinga, M. X. (2023). Simulating the Software Development Lifecycle: The Waterfall Model. *Applied* <https://doi.org/10.3390/asi6060108>
- [7] Murthy, N. M. R. (2024). Comparative Analysis of Waterfall and Agile Software Development Models: A Comprehensive Review. *International Journal of Science and Research (IJSR)*, 13(1), 580-589

Webpage

- [8] Organ, P. (2024, February 16). The hidden costs: How manual attendance tracking damages schools and disrupts parent engagement. *Orah Blog*. <https://www.orah.com/blog/hidden-costs-of-manual-attendance#:~:text=Limited%20Attendance%20Data%20Insights&text=Manual%20attendance%20tracking%20processes%20lack,patterns%2C%20and%20areas%20for%20improvement.>
- [9] Kirvan, P., Lutkevich, B., & Lewis, S. (2024, November 15). What is the waterfall model? definition and guide: Definition from TechTarget. Search Software Quality. <https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model>
- [10] Visual Paradigm. (2024). What is Unified Modeling Language (UML)? Retrieved January 1, 2025, from <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>