MOBILE APPLICATION FOR EXPLORING UKM

LI YIHAO (A197895)
Dr. Syaimak Abdul Shukor
Faculty of Information Science & Technology
Universiti Kebangsaan Malaysia
43600 Bangi, Selangor

Abstrak

Dalam beberapa tahun kebelakangan ini, dengan perkembangan pesat dalam pendigitalan pendidikan global, kebanyakan pusat pendidikan telah mula menggunakan teknologi maklumat dengan tujuan untuk mengoptimumkan pengurusan kampus mereka. Projek "Aplikasi Mudah Alih Meneroka UKM" bertujuan untuk membangunkan aplikasi mudah alih bersepadu untuk sumber asas kampus untuk menawarkan pelajar pengalaman kampus UKM yang lebih mudah. Walaupun pelajar mesti mengharungi dua cabaran untuk mengurus kehidupan akademik dan peribadi mereka dalam persekitaran kampus yang luas, kemunculan aplikasi memenuhi keperluan mendesak untuk platform bersatu yang menyediakan akses mudah kepada maklumat penting seperti laluan bas, kalendar akademik, lokasi kafeteria, berita kampus dan sistem hal ehwal pendidikan. Aplikasi ini direka bentuk melalui reka bentuk berpusatkan pengguna, dicapai melalui tinjauan dan maklum balas yang digabungkan ke dalam pembangunan berulang, untuk meringankan sedikit kehidupan pelajar dan mengurangkan tekanan sambil menguruskan masa dengan lebih berkesan. Projek ini menggunakan Kotlin dalam pembangunannya, menggunakan amalan terbaik semasa dalam pengaturcaraan untuk mencipta antara muka yang teguh dan mesra pengguna. Perkembangan masa depan mungkin melibatkan kemas kini masa nyata dan kemasukan ciri yang diminta oleh pengguna untuk memastikan aplikasi kekal sebagai alat yang berharga, berkembang mengikut keperluan komuniti UKM. Inisiatif ini akhirnya berhasrat untuk menyokong pelajar dalam meneruskan perjalanan akademik mereka dan memaksimumkan pengalaman kampus mereka.

Kata kunci: Aplikasi Mudah Alih, Sumber Kampus, Reka Bentuk Berpusatkan Pengguna

Abstract

In the last couple of years, with rapid development in global education digitization, most education centres have started using information technology with a view to be optimizing the management of their campuses. The “Mobile Application for Exploring UKM” project aims to develop an integrated mobile application for campus basic resources to offer students a more convenient UKM campus experience. While students must navigate through dual challenges of managing their academic and personal lives in a sprawling campus environment, the emergence of the app meets the urgent need for a unified platform that provides easy access to crucial information such as bus routes, academic calendars, cafeteria locations, campus news and educational affairs systems. This application is designed through user-cantered design, achieved by surveys and feedback incorporated into iterative development, to lighten the student's life a bit and reduce stress while managing time more effectively. The project uses Kotlin in its development, applying the current best practices in programming to create a robust and user-friendly interface. Future developments might involve real-time updates and the inclusion of features requested by users to make sure the app remains a valuable tool, evolving with the needs of the UKM community. This initiative ultimately intends to support students in pursuing their academic journey and maximizing their campus experience.

Keywords: Mobile Apps, Campus Resources, User-Cantered Design

1. Introduction

In recent years, the rapid advancement of mobile technology has transformed the way individuals access information, communicate, and manage daily activities. Within the context of higher education, this transformation has created new opportunities for improving student experience through digital platforms. At large and complex institutions such as Universiti Kebangsaan Malaysia (UKM), students face the challenge of navigating a wide range of academic, administrative, and campus life resources that are often scattered across multiple systems. This fragmentation can lead to inefficiencies, wasted time, and unnecessary stress. Recognizing these challenges, the “Mobile Application for Exploring UKM” project was conceptualized to centralize essential campus resources into a single, user-friendly mobile platform. Designed primarily for Android devices, the application aims to provide UKM students with efficient access to academic tools, transportation information, dining options, and campus updates, thereby improving convenience and overall quality of campus life.

The proposed application consolidates multiple services into one cohesive system, addressing the inefficiencies of existing fragmented information channels. Its core functions include bus

route and schedule tracking, an academic calendar for important dates and events, a dining locator integrated with navigation assistance, and quick links to UKM's academic platforms such as UKMFOLIO and SMPWEB. Additionally, a UKM news section ensures that students remain informed about the latest announcements and developments on campus. To enhance personalization, the app includes a secure login and registration system, enabling students to create accounts for tailored access. An administrator portal allows authorized personnel to manage and update core features, ensuring that information remains accurate and relevant. By adopting a centralized approach, the application eliminates the need for students to repeatedly search across multiple platforms, fostering a more connected, organized, and stress-free university experience.

The significance of this project extends beyond convenience, as it also supports efficiency, accessibility, and institutional modernization. By integrating key services into one mobile application, UKM can enhance operational effectiveness while providing students with a reliable, technology-driven support tool. The project draws inspiration from successful university mobile solutions, adapting their best practices to suit UKM's unique requirements. Leveraging Android Studio and Kotlin for development ensures compatibility, maintainability, and scalability. Furthermore, the structured adoption of the Waterfall development model provides clear stages for requirement gathering, design, implementation, testing, and maintenance, ensuring systematic progress and quality assurance. Ultimately, the "Mobile Application for Exploring UKM" serves as a forward-looking initiative that not only addresses immediate student needs but also lays the foundation for future expansion, including real-time updates, enhanced interactivity, and continuous improvement based on user feedback.

2. Literature Review

The advancement of mobile internet technology has significantly influenced information management and service delivery within higher education institutions. Traditional approaches often require students to access multiple, fragmented systems for tasks such as academic websites, learning platforms, bus schedules, and dining information. This decentralization can lead to inefficiency and information omission. Universities worldwide have begun addressing this by developing centralized mobile applications that integrate key campus services into a single platform. Such integration not only improves accessibility and efficiency but also supports sustainability goals by reducing reliance on paper-based resources. For Universiti Kebangsaan Malaysia (UKM), the development of a "Mobile Application for Exploring UKM" aims to address the specific needs of students, particularly in streamlining access to academic,

transportation, and campus life information.

Several existing solutions provide valuable insights into effective campus mobile application design. The MYUKM app, developed by UKM, focuses on three main features—Smart Attendance using QR codes, Smart Bus with real-time tracking, and centralized UKM news updates. MYLBORO, created by Loughborough University, offers personalized schedules, geolocation-based attendance check-in, and integration with library and academic systems. The MIT Mobile application, developed by the Massachusetts Institute of Technology, incorporates advanced features such as emergency and maintenance reporting, offline functionality, and comprehensive student support services. Comparative analysis shows that while MYUKM emphasizes essential operational efficiency, MYLBORO balances academic and personal resource integration, and MIT Mobile prioritizes broad functionality and resilience, even in low-connectivity environments. Each demonstrates unique strengths in performance, usability, and scope, reflecting different institutional priorities and technical approaches.

Building on these studies, the proposed UKM mobile application will adopt a modular and integrated design that consolidates essential campus functions such as bus routes, campus mapping, academic calendar updates, dining locations, and direct access to SMPWEB and UKMFOLIO. Inspired by the user-friendly designs of MYLBORO and MIT Mobile, the interface will prioritize clarity, simplicity, and ease of navigation. Real-time synchronization with official UKM data sources will ensure accuracy, while an in-app feedback mechanism will enable continuous improvement based on user needs. By combining functional breadth with usability and reliability, the proposed solution seeks to provide UKM students, especially new and international students, with a unified, efficient, and trustworthy platform. This approach not only addresses current gaps in campus information delivery but also supports UKM's long-term vision for enhanced digital campus services.

3. Methodology

The development of the "Mobile Application for Exploring UKM" followed the Waterfall Model, a sequential software development methodology that clearly defines each stage before progressing to the next. This model was selected for its structured approach, which supports thorough planning, precise design, and systematic testing. The process began with Requirement Analysis, involving surveys and feedback from UKM students to identify essential features such as bus route tracking, academic calendar access, campus map navigation, and quick links to academic platforms. The System Design stage then translated these requirements into detailed interface layouts and database structures, ensuring both usability and scalability.

In the Implementation phase, the application was developed using Kotlin in Android Studio, with SQLite as the primary database. Functional modules—such as registration/login, bus route queries, event reminders, and café locators—were coded incrementally. Continuous testing was integrated into development to detect and fix errors early. The Verification Testing stage applied rigorous functional and non-functional testing, including performance checks under low connectivity, to ensure stability. Finally, Maintenance processes were planned to incorporate user feedback, deliver periodic updates, and expand functionality while maintaining system performance.

To support efficient operation, the system adopts a read-write separation architecture, with a primary server for write operations and secondary servers for read requests, optimizing data access during peak usage. The application meets quality requirements by offering an intuitive interface, robust encryption for user data, and modular scalability for future enhancements. Hardware and software requirements were carefully defined for both developers and users to ensure smooth performance across devices. This methodology ensured that the final product is not only functionally complete but also user-friendly, secure, and adaptable—providing UKM students and staff with a reliable, centralized platform for campus exploration and information access.

4. Results

4.1 Application Development

This chapter systematically elaborates on the development and testing process of " Mobile Application for Exploring UKM ". The system is implemented using Android Studio and Kotlin language, combined with Jetpack Compose to build the user interface, and uses Room database for local data storage and management. The modules cover user login and registration, personal profile, UKM website links, bus routes, academic calendars, news activities, and restaurant navigation, and design different functional layouts for the student end and the administrator end to ensure the modularization and maintainability of the system architecture.

The testing stage adopts the black box testing method, including equivalence class partitioning, boundary value analysis, decision table testing, and state transition testing. A requirement traceability matrix was established, covering both functional and non-functional tests, to ensure that input validation, link validity, data security, and system compatibility all meet the standards. The results show that all core modules run stably under normal and boundary conditions, meeting the design requirements. This chapter finally confirms that the system has complete functions, friendly interaction, and good operational performance.

4.1.1 Homepage

Figure 4.1 shows the UI design of the student homepage, which adopts a simple card-style layout to present the core function entrances. At the top of the page, the student's basic information (class/ID number) is displayed, and below, there are six functional modules such as website navigation, personal profile, bus route, calendar, etc. At the bottom, there is a logout button.

Figure 4.1 Home page

4.1.2 UKM website page

Figure 4.2 illustrates the content layout of the UKM university website page, featuring three main campus system quick access entries. Each entry is clearly labelled with the system name (such as UKM FOLIO), the website link, and the function description. The layout is grouped to enhance the efficiency of information acquisition.

Figure 4.2 UKM website page

4.1.3 Profile page

Figure 4.3 shows the layout design of the student profile page, which presents the student's academic information and contact details in a clear group structure. The top of the page prominently displays the student's name, class, and student number, and below this, the 9 detailed items such as phone number, academic year, and major are listed in sequence. Important data (matric and email) are displayed in a fixed format.

Figure 4.3 Profile page

4.1.4 Bus route page

Figure 4.4 shows the layout of the campus bus route page, presenting the schedules and routes of each bus stop in a clear and organized manner using date labels and categorized lists. At the top of the page, there is an option to switch days. Below, the different areas display the departure times, starting and ending stops of the buses, and a "Details" button is provided to expand the functionality.

Figure 4.4 Bus route page

4.1.5 Calendar page

Figure 4.5 presents the design of the calendar page, which is presented in a tabular form to clearly display the date distribution for November 2024. It also marks the attendance days and holidays with different colors. The top of the page highlights the month and year, and below, a weekly calendar table shows the complete dates. Important dates (such as holidays) have special visual indicators.

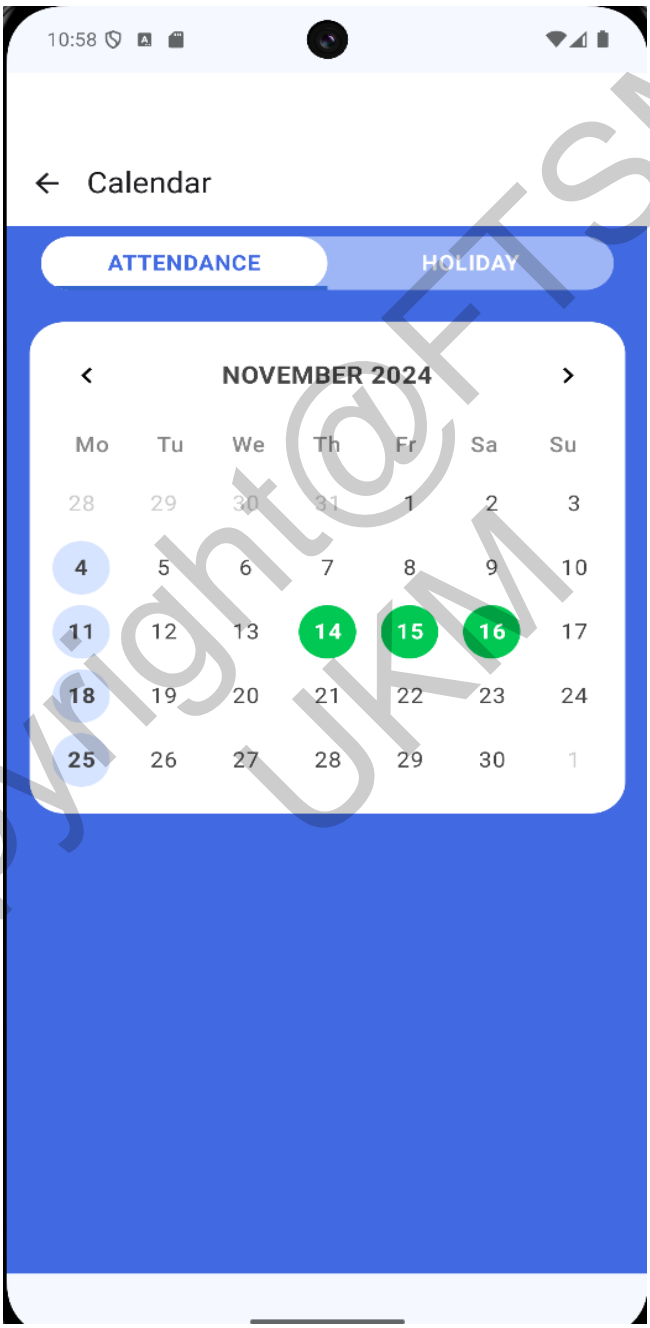

Figure 4.5 Calendar page

4.1.6 Events and News page

Figure 4.6 presents the design of the campus activities and news pages, adopting a card-style layout to clearly display three important campus announcements (Open Day, Academic Seminar, and Final Examination Notice). Each announcement includes the event title, date and time, and a brief description. Visual segmentation is achieved through dividing lines, and the important time information is displayed in bold.

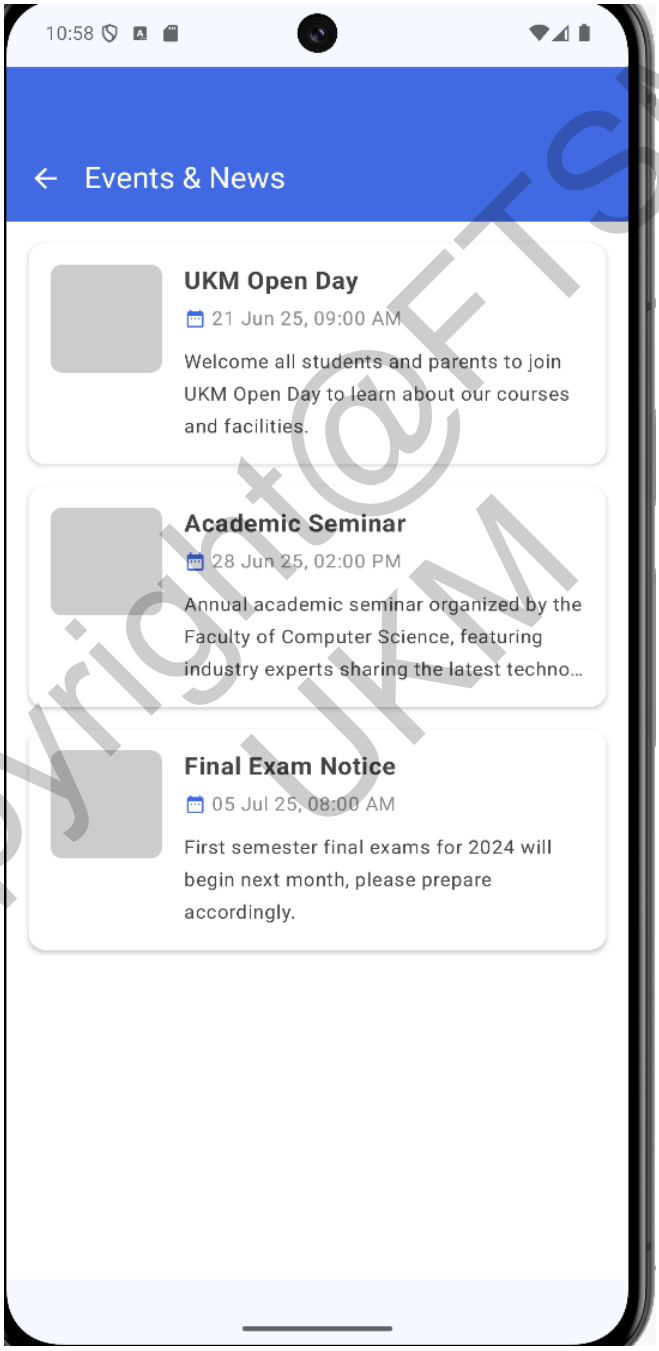

Figure 4.6 Events and News page

4.1.7 Café locator page

Figure 4.7 shows the design of the campus café positioning page, which presents the business hours, locations and feature descriptions of each café in a card layout. The page includes a "Navigate to Location" function button and uses checkboxes to mark specific location information, providing intuitive spatial positioning guidance.

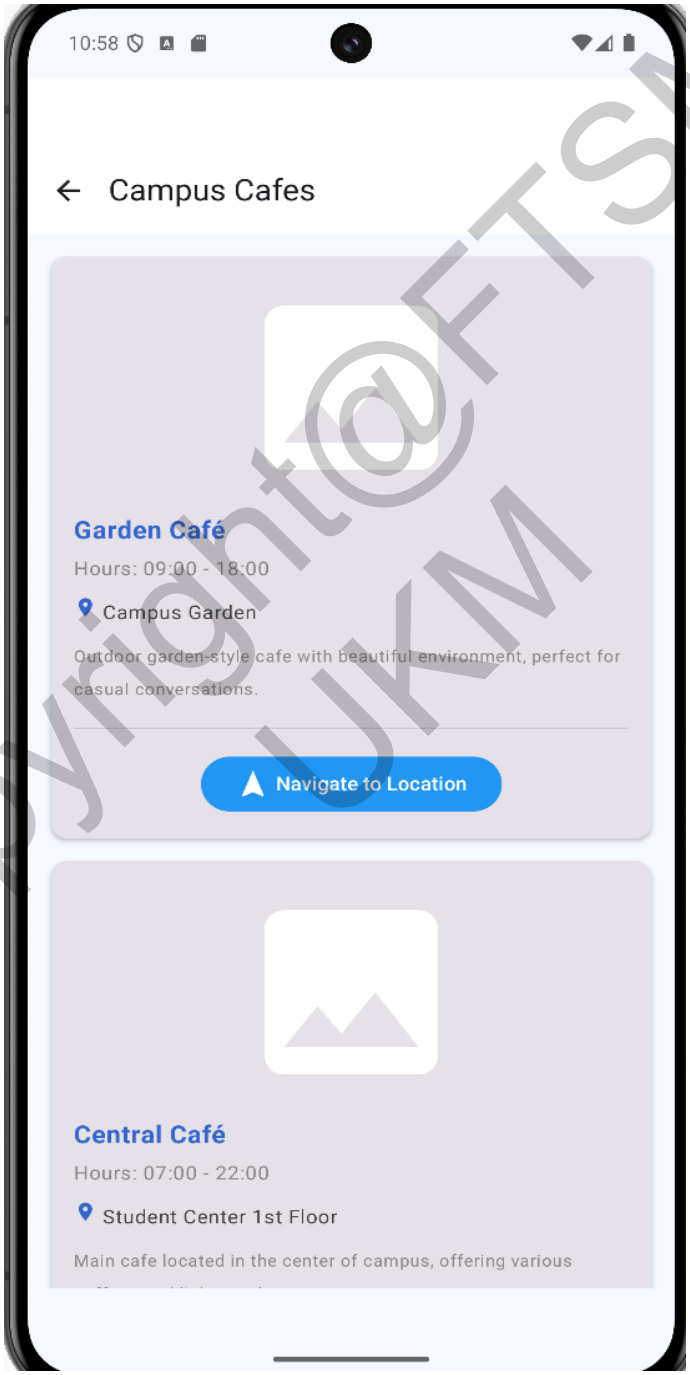

Figure 4.7 Café locator page

4.1.8 Administrator homepage

Figure 4.8 shows the interface design of the administrator's homepage. It adopts a modular layout to present the four major management functions: website, calendar, news, and café management. The top of the page displays the administrator's welcome message and identity identifier, while below, the entry points and brief descriptions of each management module are clearly presented through a categorized list.

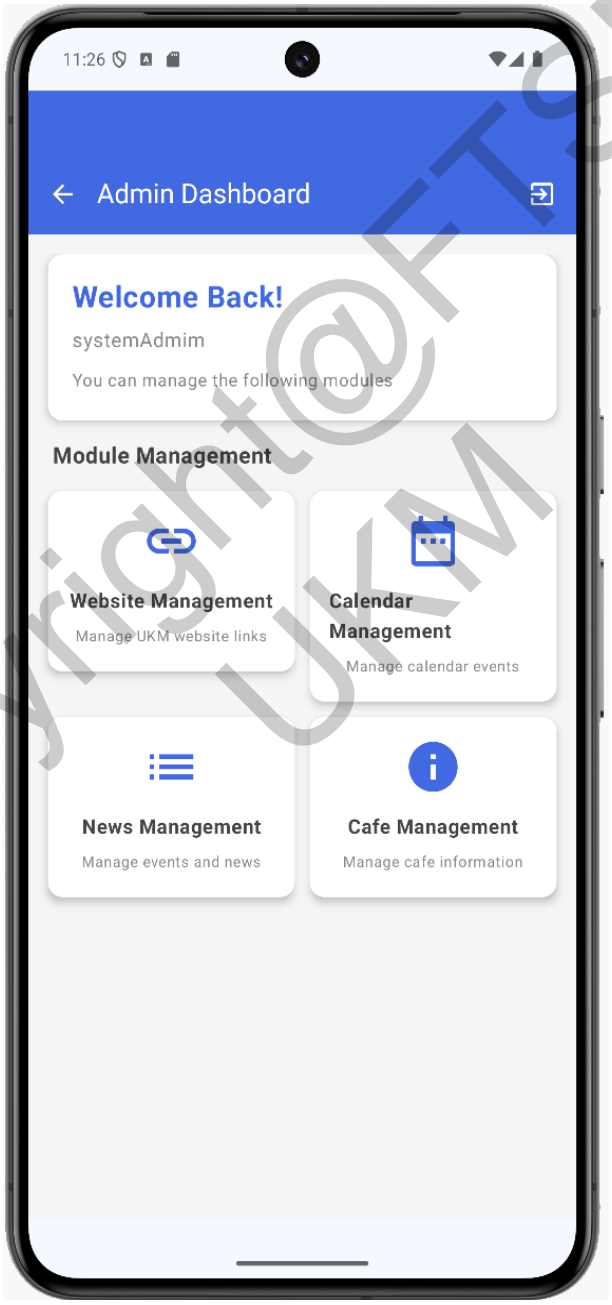

Figure 4.8 Administrator homepage

4.1.9 Website Management Page

Figure 4.9 shows the design of the website management page, which adopts a layered layout to display existing website information (such as the UKM FOLIO system) and editing forms. The page provides editing functions for the three fields of title, URL and description, and has a cancel/save operation button at the bottom. It supports administrators to maintain and update the links of the campus website.

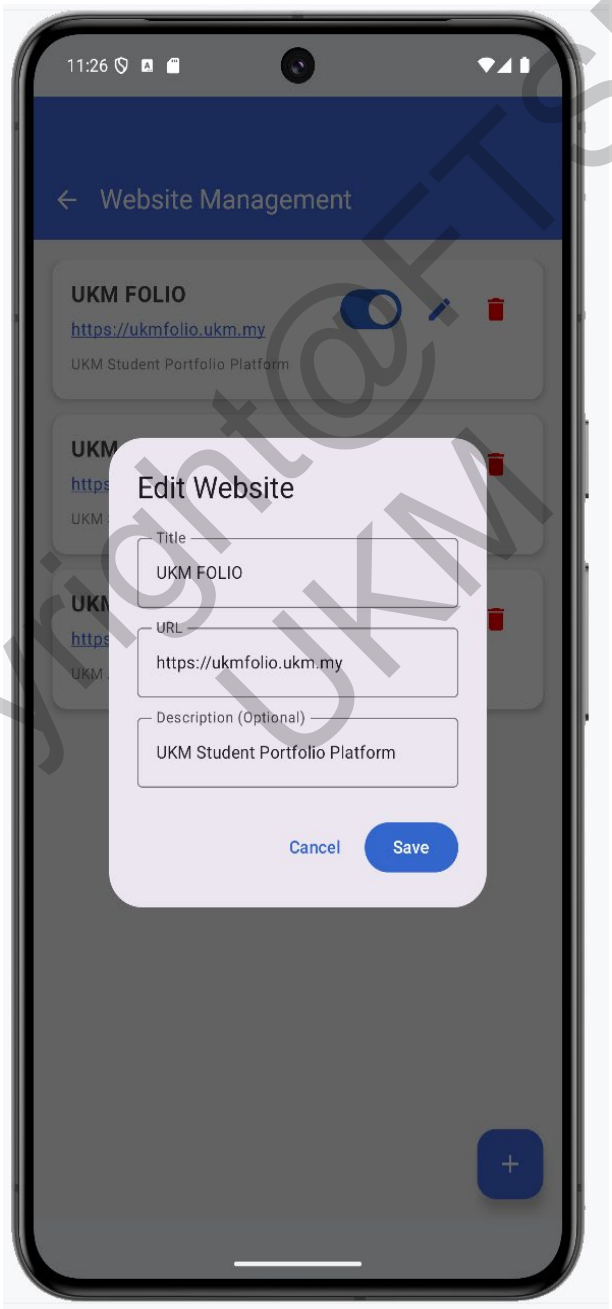

Figure 4.9 Website Management Page

4.1.10 Calendar Management Page

Figure 4.10 shows the design of the calendar management page, which adopts an event card layout to display important dates such as holidays and exam days along with their type labels. Each event includes the date, category (Holiday/Exam), and detailed description, enabling administrators to systematically manage and maintain the campus schedule.

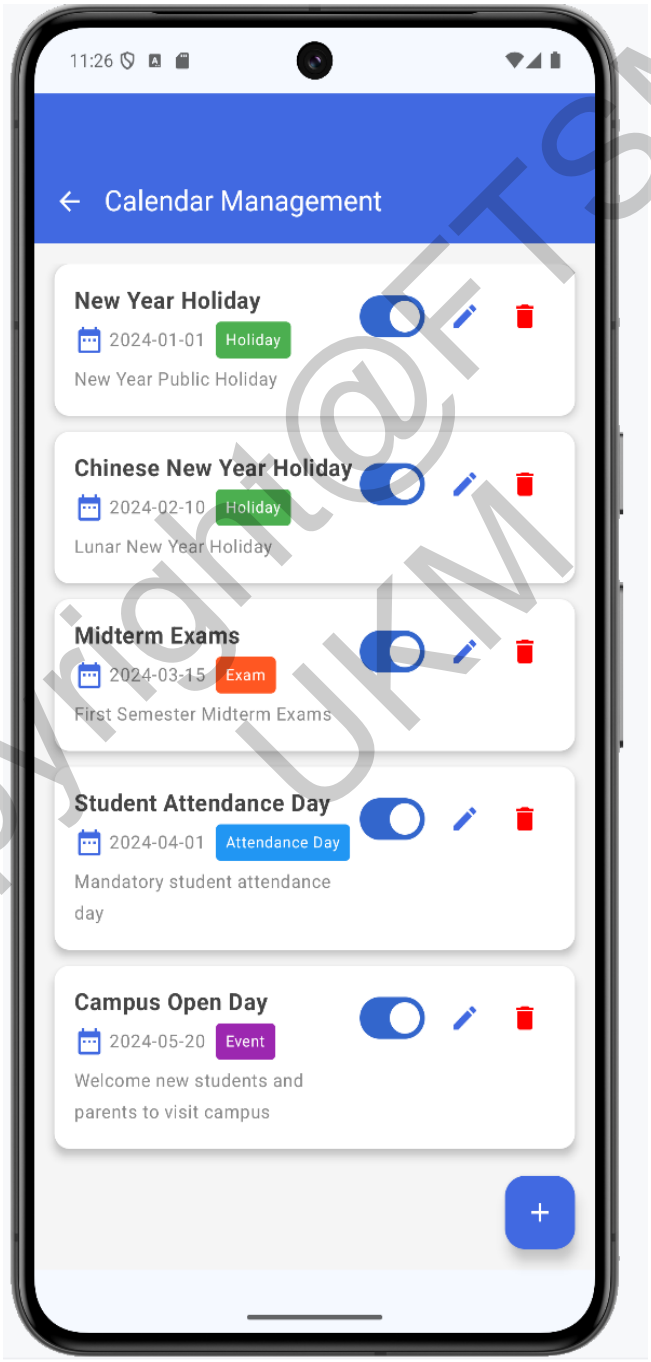

Figure 4.10 Calendar Management Page

4.1.11 News Management Page

Figure 4.11 shows the design of the news management page, which presents campus event announcements in a card-style layout, including the time, location and details. The page enables administrators to uniformly edit and publish important event information such as open days and academic lectures.

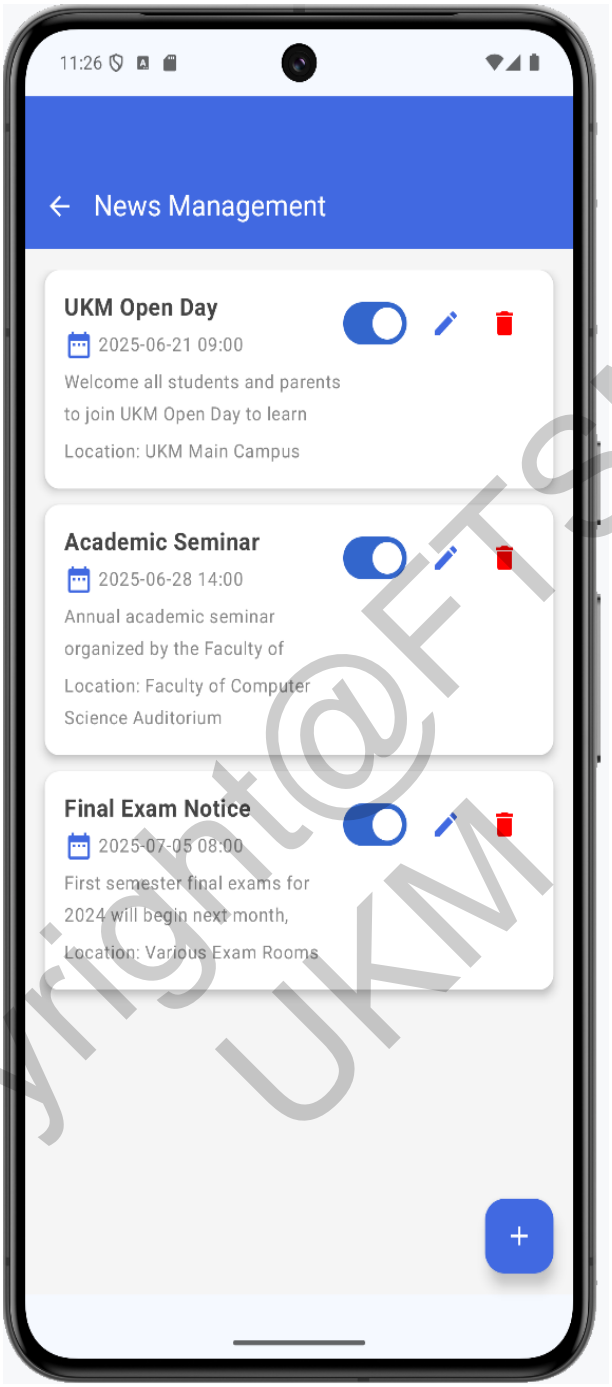

Figure 4.11 News Management Page

4.1.12 Cafe Management Page

Figure 4.12 shows the design of the café management page, which presents the business hours, locations and feature descriptions of each café in a card layout. The page enables the administrator to uniformly maintain and manage the basic information of multiple cafés on campus.

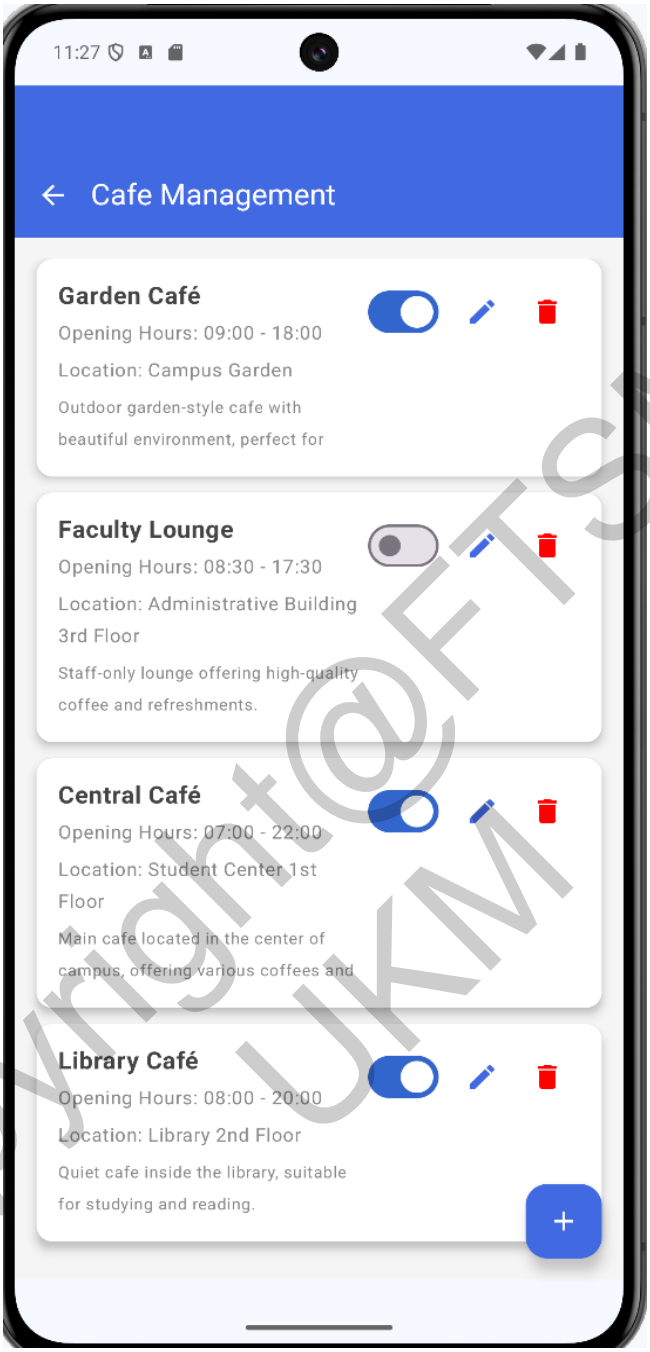

Figure 4.12 Cafe Management Page

4.2Testing

4.2.1 Traceability Matrix Testing

The traceability matrix ensures that every functional requirement specified during the development phase is at least covered by one corresponding test case. This enables the development team to track the test coverage and confirm that no critical functions are missing. As shown in the table above, all major functions, such as user authentication, bus route tracking,

calendar access, café navigation, and news updates, have dedicated test cases. This approach enhances the completeness of the tests and supports defect tracking for the system during the testing phase.

Table 4.1 Traceability Matrix Testing

Requirement ID	Functional Requirement Description	Test Case ID	Test Case Description
FR1.0	Users can register an account (enter student number, email address and password)	TC01	Test the successful registration process for new users
FR1.1	Verify whether the student number is valid during registration	TC02	When an invalid student number is entered, the system prompts an error message
FR2.1	Users can log in to the system after entering the correct information	TC03	Log in to the system with the correct student number and password
FR2.2	When a user enters incorrect login information, the system gives a prompt	TC04	When attempting to log in with the wrong password, the system returns an error prompt
FR3.1	Users can access the UKM school bus route function	TC05	Go to the "Bus Route" page to display the available routes
FR3.2	Check the detailed school bus stops and schedules	TC06	Select a certain route, and the system will display detailed stops and departure times
FR4.1	Users view the academic calendar and the schedule of on-campus activities	TC07	Open the Calendar module to display academic schedules and vacations
FR4.2	Users can set reminders	TC08	Set important exam reminders and verify whether the reminders have been successfully saved
FR5.1	Provides quick links to UKMFOLIO and SMP platforms continuation	TC09	Click the link to jump to the UKM education system and verify whether the jump is correct to be continued
FR6.1	Users can view the information and business hours of the campus coffee shop	TC10	Display the name, description and business hours of the coffee shop

FR6.2	Use Google Maps to navigate to the restaurant	TC11	Click the navigation button to jump to Google Maps and display the target location
FR7.1	Users view the latest campus news and announcements	TC12	Open the News module to display the latest event information and announcements

4.2.2 Functions Testing

Tables 4.2 and 4.3 show the systematically categorized functional (FR) and non-functional (NFR) testing requirements for the UKM campus App, clearly marking the risk levels (high/medium/low) for each module. The functional tests cover the core user operations (login/ bus query, etc.), while the non-functional tests focus on performance optimization (loading speed/compatibility) and data security (encryption/permissions), jointly forming a complete quality assurance system.

Table 4.2 Functional Testing

Function ID	Function Description	Risk Level
FR1.0	User registration: Enter your student number, email address and password to create an account	Low
FR2.1	User registration: Enter your student number, email address and password to create an account	High
FR3.1	School bus route inquiry: Browse and select UKM school bus route information	Low
FR3.2	Display specific sites and schedules	High
FR4.1	Display specific sites and schedules	Low
FR4.2	Set academic reminders and event markers	Low
FR5.1	Quick links to visit UKMFOLIO and SMP websites	High
FR6.1	Browse the information of the campus cafe (name, time, description)	Low
FR6.2	Call Google Maps to navigate to the restaurant	Middle
FR7.1	Check campus news and event announcements	Low

Table 4.3 Non-functional Testing

Function ID	**Function Description**	**Risk Level**
NFR1	Compatibility testing	Low
BFR2	Stability test	High
NFR3	Authority management	Low

4.2.3 Testing Techniques

In testing the Mobile Application for Exploring UKM, black box testing is adopted as the

primary testing method for validating system functionality. This approach focuses on evaluating the app's behaviour and outputs based on input conditions, without concern for the internal code or logic implementation. Since the application is primarily developed for Android and involves multiple user interface modules such as login, calendar, bus routes, and news display, black box testing is particularly effective for verifying these external behaviours. The following black box testing techniques were applied:

Equivalence Partitioning (EQ)

Inputs are divided into valid and invalid partitions. For example, during the registration module, user input fields such as student ID, email, and password are tested using valid and invalid formats to ensure the system handles them appropriately.

Boundary Value Analysis (BVA)

Focuses on the values at the edges of input ranges. For instance, when testing the calendar reminder, boundary values such as the first day of the semester or the last available date are used to validate correct behavior.

Decision Table Testing (DTT)

Used in scenarios with multiple input conditions. For example, login functionality requires both a valid student ID and password. Various combinations of valid/invalid inputs are used to ensure correct error handling.

State Transition Testing (STT)

Applied in modules where system response depends on current state and input events. For instance, when a user navigates from the home page to the bus route module and then back, the system's state should update correctly.

Use Case Testing (UCT)

Based on functional use cases defined during the design phase. Each use case, such as "Viewing News" or "Accessing UKMFolio", is tested through direct user interaction simulations to ensure the app responds correctly.

4.2.4 Testing Activities

Table 4.4 shows the list of tasks that need to be completed to test the information of application.

Table 4.4 Non-Working Testing Tested Table

No.	Task Activity	Description

1	Test Planning and Control	Define the test scope, objectives, schedule, and identify modules to be tested.
2	Test Design and Analysis	Review system specifications and requirements to design relevant test cases.
3	Testing Environment Setup	Prepare the necessary software (Android Studio) and hardware (Android devices).
4	Test Implementation	Execute test cases, apply black-box techniques, and record outcomes.
5	Documentation of Testing Summary	Compile a testing summary report including results, pass/fail rates, and issues.
6	Documentation and Closure	Ensure all test documents are completed, reviewed, and officially submitted.
7	Final Test Completion	Confirm that all planned tests have been executed and all defects addressed.

4.2.5 Design Between Test Cases

In designing test cases for the Mobile Application for Exploring UKM, a structured approach was adopted to ensure coverage of all critical system functions while minimizing redundancy. The test case design followed the principle of modular separation, meaning each test case corresponds to a specific feature or sub-feature (e.g., login, bus route, calendar, café locator). This modular design helps in isolating defects and tracing issues directly back to the associated requirement or module. To ensure logical coherence between test cases, dependencies and flow relationships were carefully considered. For instance, the login test case (TC01) must be executed successfully before proceeding to test cases involving authenticated access, such as calendar viewing (TC07) or feedback submission (TC13). This sequence maintains real-world usage patterns and validates proper state transitions within the app.

Moreover, the test cases are designed to account for both positive paths and negative paths. Positive path test cases validate that the system functions correctly with valid inputs (e.g., successful login, proper bus route retrieval), while negative path cases evaluate how the system handles invalid inputs or unexpected behaviours (e.g., empty input fields, invalid dates, no internet connection). In between these cases, transition testing (state testing) was applied to ensure that moving from one module to another (such as from homepage to café locator) retains session state, user data, and interface consistency. Each test case was also assigned a unique identifier and linked with the traceability matrix, ensuring that every functional requirement (FR) is validated by at least one corresponding test case. This interconnected design between test cases supports effective regression testing and enhances defect tracking throughout the test cycle.

4.2.6 Account Register Testing

Four methods were used to test the user registration module. The following is the detailed test

table.

(1) Equivalence Partitioning

The input is divided into valid and invalid equivalence classes. Each test case represents a typical class, thereby reducing the number of tests while covering the key types. Table 4.5 illustrates the testing process of the Equivalence Partitioning method.

Table 4.5 Equivalence Partitioning

Test Case ID	Input Condition	Equivalence Class	Expected Result
EQ01	Valid ID: A123456, Email: abc@ukm.edu.my, Password: pass123	Valid	Registration success
EQ02	Invalid ID: 123, Email: abc@ukm.edu.my, Password: pass123	Invalid ID	Registration failed (ID)
EQ03	ID: A123456, Email: abc.ukm.edu.my, Password: pass123	Invalid Email	Registration failed (Email)
EQ04	ID: A123456, Email: abc@ukm.edu.my, Password: 123	Invalid Password	Registration failed (Password)

(2) Boundary Value Analysis

Test the boundary values of password and student ID input lengths. Whether the 6-digit password boundary is accepted. Table 4.6 shows the testing process of the Boundary Value Analysis method.

Table 4.6 Boundary Value Analysis

Test Case ID	Field	Input Value	Expected Result
BVA01	Password	5 characters: "abc12"	Registration failed (too short)
BVA02	Password	6 characters: "abc123"	Registration success
BVA03	Student ID	7 characters: "A123456"	Registration success
BVA04	Student ID	0 chars (empty)	Registration failed (empty ID)

(3) Decision Table Testing

Using four combinations of conditions to establish a decision table to show how the input combinations affect the registration result, and to facilitate the coverage of all logical paths. Table 4.7 shows the testing process of the Decision Table Testing method.

Table 4.7 Decision Table Testing

Condition/Input	DTT01	DTT02	DTT03	DTT04
Student ID is valid	Y	Y	N	Y
Email format is valid	Y	N	Y	Y
Password length ≥ 6	Y	Y	Y	N
Student ID already exists	N	N	N	N
Expected Result	Success	Fail (Email)	Fail (ID)	Fail (Password)

Note:

Y: YES

N: NO

(4) Use Case Testing Table

Based on the user's usage process, design test scenarios that closely resemble real operations. Check whether the system's feedback under the user's actions conforms to the requirements and user experience expectations. Table 4.8 illustrates the testing process of the Use Case Testing Table method.

Table 4.8 Use Case Testing

Use Case ID	Scenario Description	Expected Result
UCT01	User fills all fields correctly and clicks "Register"	Successful registration; redirect to login
UCT02	User enters existing email and ID	Registration fails; error "ID/Email exists"
UCT03	User leaves email blank	Registration fails; prompt "Email required"
UCT04	User inputs invalid ID format, valid others	Registration fails; prompt "Invalid ID format"

(5) Conclusion

These four testing techniques help ensure the completeness and accuracy of account registration testing. Equivalence partitioning reduces test quantity while ensuring coverage of input types. Boundary value analysis focuses on common input edge errors, especially for password validation. Decision tables help reveal logic flaws under condition combinations. Use case testing simulates real user behaviour and validates expected system responses. Together, these techniques ensure that the registration module

is reliable.

4.2.7 Test result printing and status verification

State transition testing will be used for the design of the test of the application:

State transition testing

Table 4.9 and 4.10 shows the state transition information and test coverage details of "Mobile Application for Exploring UKM"

Table 4.9 State Information

Test Coverage ID	Recent Condition	Activity Action	Next Condition
ST01	App launched	Navigate to Login Page	Login Page
ST02	On Login Page	click login	Authenticated Home Page
ST03	User Logged In	Click on Bus Route	Bus Route Page
ST04	On Home Page	Click Logout	Login Page
ST05	On Register Page	Submit registration form	Redirect to Login Page

Table 4.10 Test Coverage

Test Coverage ID	Recent Condition	Activity Action	Next Condition
TCOV01	Login Page	Submit valid credentials	Home Page
TCOV02	Home Page	Click Café locator	Café Locator Page
TCOV03	Bus Route Page	Return to Home	Home Page
TCOV04	Register Page	Submit valid registration	Login Page
TCOV05	Profile Page	Click Logout	Login Page

Table 4.11 illustrates the change paths of each significant state of the application.

Table 4.11 State Information

Test Coverage ID	Recent Condition	Activity Action	Next Condition
COV01	User launches app	Show splash screen	User not logged in
COV02	User not logged in	Show login/register page	User clicks Register

COV03	User clicks Register	Display registration form		User enters valid info
COV04	User enters valid info	Submit registration form		Registration successful
COV05	Registration successful	Redirect to login page		User logs in

Table 4.12 shows the test coverage for the system.

Table 4.12 Test Coverage for the application

Test Condition ID	Test Coverage (Coverage ID)
TCON01	COV01
TCON02	COV02
TCON03	COV03
TCON04	COV04
TCON05	COV05

4.2.8 Login Testing

Equivalence Partitioning (EQ)

Table 4.13 presents the effective/invalid login inputs and the expected results.

Table 4.13 Equivalence Partitioning for Login Testing

Test Case ID	Input Condition (Student ID / Password)	Equivalence Class	Expected Result
EQ-L01	Valid ID (A123456) / Valid Password (pass123)	Valid Inputs	Login Success
EQ-L02	Invalid ID (123) / Valid Password (pass123)	Invalid ID	Error: "Invalid ID"
EQ-L03	Valid ID (A123456) / Invalid Password (123)	Invalid Password	Error: "Invalid Password"
EQ-L04	Empty ID / Empty Password	Empty Fields	Error: "Fields Required"

Boundary Value Analysis (BVA)

Table 4.14 shows the tests for password/account length restrictions for login.

Table 4.14 Boundary Value Analysis for Login Testing

Test Case ID	Field	Input Value	Expected Result
BVA-L01	Password	5 chars ("Abcde")	Error: "Password too short"
BVA-L02	Password	6 chars ("Abcde")	Error: "Invalid ID format"
BVA-L03	Student ID	7 chars ("A123456")	Login Success

BVA-L04	Student ID	6 chars ("a12345")	Error: "Invalid ID format"

Decision Table Testing (DTT)

Table 4.15 demonstrates the logical combination of account/password validity.

Table 4.15 Decision Table for Login Testing

Condition/Input	DTT-L01	DTT-L02	DTT-L03	DTT-L04
Valid Student ID	Y	Y	N	N
Valid Password	Y	N	Y	N
Account Exists in database	Y	Y	N	N
Expected Result	Y	Y	N	N

Note:

Y: Yes (Valid/Exists)

N: No (Invalid/Does Not Exist)

State Transition Testing (STT)

Table 4.1 demonstrates the logical combination of account/password validity.

Table 4.16 State Transition Testing for Login

Test Case ID	Current State	Action	Next State	Expected Output
STT-L01	App Launched	Navigate to Login	Login Page	Login UI Loaded
STT-L02	Login Page	Enter Valid Credentials	Home Page	Redirect to Home
STT-L03	Login Page	Enter Invalid Credentials	Login Page	Error Message
STT-L04	Home Page	Click Logout	Login Page	Session Cleared

4.2.9 UKM Website Function Testing

Equivalence Partitioning (EQ)

Table 4.17 shows the test results of valid/invalid/empty/damaged links to verify the correct redirection or error handling during access to the UKM website.

Table 4.17 Equivalence Partitioning for UKM Website Testing

Test Case ID	Input Condition (Website Link)	Equivalence Class	Expected Result
EQ-W01	Valid Link	Valid	Redirects to UKMFOLIO

EQ-W02	Invalid Link	Invalid	Error: "Invalid Link"
EQ-W03	Empty Link	Empty	Error: "Link Required"
EQ-W04	Broken Link	Invalid	Error: "Page Not Found"

Boundary Value Analysis (BVA)

Table 4.18 presents the extreme values for checking the URL length/protocol (1 - 256 characters, without "https://") to verify the system limitations and protocol execution.

Table 4.18 Boundary Value Analysis for UKM Website Testing

Test Case ID	Field	Input Value	Expected Result
BVA-W01	URL Length	1 character ("a")	Error: "Invalid Link"
BVA-W02	URL Length	255 characters (max valid URL)	Redirects successfully
BVA-W03	URL Length	256 characters (exceeds limit)	Error: "Link Too Long"
BVA-W04	URL Protocol	Missing "https://"	Missing "https://"

Decision Table Testing (DTT)

Table 4.19 shows the factors such as evaluating the validity of the link, checking if the database exists, and verifying the login status, to ensure that the correct response of success or failure can be given.

Table 4.19 Decision Table for UKM Website Testing

Condition/Input	DTT-W01	DTT-W02	DTT-W03	DTT-W04
Link Format Valid	Y	Y	N	N
Link Exists in Database	Y	N	Y	N
User Logged In	Y	Y	Y	N
Expected Result	Success	Error	Error	Error

Note:

Y: Yes (Valid/Exists)

N: No (Invalid/Does Not Exist)

State Transition Testing (STT)

Table 4.20 presents through the map user flowchart that it is clearly visible how the user enters an external browser from the home page and then returns to the home page. This helps to confirm the smoothness of the transition process and the error recovery mechanism.

Table 4.20 State Transition Testing for UKM Website

Test Case ID	Current State	Action	Next State	Expected Output
STT-W01	Home Page	Click "UKM Website" Button	Website List	Displays Links
STT-W02	Home Page	Select Valid Link	External Browser	Opens UKMFOLIO
STT-W03	Home Page	Select Invalid Link	Website List	Error Message
STT-W04	External Browser	Click "Back" Button	Website List	Returns to App

4.2.10Specifications of Testing Procedures

The Specifications of Testing Procedures in the Mobile Application for Exploring UKM define structured test plans and ensure each core module like login, calendar, bus routes, and news－is thoroughly validated. These specifications include the Test Procedure ID, testing objectives, defined test cases, and the step-by-step procedures to execute each test. The goal is to verify system functionality, stability, and user interface usability through systematic black-box methods such as equivalence partitioning, boundary value analysis, and use-case testing. These procedures ensure that each function meets the expected behaviour and maintains consistency across different user interactions and usage conditions.

Table 4.21 shows the information about the application.

Table 4.21 Testing Procedures

Test Procedure ID	Objective	Test Case	Procedure
TP01	Verify successful registration	TC01	Enter valid student ID, email, password → Click register → Expect success

TP02	Validate ID during registration	TC02	Enter invalid student ID → Click register → Expect error prompt
TP03	Confirm successful login	TC03	Enter valid credentials → Click login → Redirect to homepage
TP04	Detect login error	TC04	Enter invalid password → Click login → Expect error message
TP05	Display available bus routes	TC05	Log in → Navigate to Bus Route → Expect route list to load
TP06	Show detailed bus stop and timing	TC06	Select a route → Check stops info and time
TP07	Show academic calendar	TC07	Log in → Open Calendar → Expect academic events to display
TP08	Set reminder for an event	TC08	Add reminder in calendar → Save → Confirm success message

to be continued

continuation

TP09	Open UKMFOLIO via quick link	TC09	Click educational link → Redirect to UKM platform
TP10	Display café information	TC10	Navigate to café module → Expect café name, description, hours
TP11	Navigate café using Google Maps	TC11	Tap navigate → Launch Google Maps with café location
TP12	Display UKM announcements	TC12	Navigate to News → Check if latest announcements are listed

4.2.1 Testing Result

The Testing Result section summarizes the outcomes of executing the defined test cases. Each result is documented under a specific Test Case ID and linked to its associated procedure. The mobile app was tested manually using Android Studio emulators and real devices. All modules include registration, login, navigation, calendar, and news were evaluated for functionality, input validation, and integration reliability. The results were consistent with expected results, and all tests successfully passed. The results confirm that the application performs reliably across all critical modules and is ready for deployment, offering UKM students seamless and functional user experience.

Table 4.22 Testing Result Table

Test Case ID	Test Procedure ID	Tool	Pass/Fail
TC01	TP01	Manual	Pass
TC02	TP02	Manual	Pass
TC03	TP03	Manual	Pass
TC04	TP04	Manual	Pass
TC05	TP05	Manual	Pass
TC06	TP06	Manual	Pass
TC07	TP07	Manual	Pass
TC08	TP08	Manual	Pass
TC09	TP09	Manual	Pass
TC10	TP10	Manual	Pass
TC11	TP11	Manual	Pass
TC12	TP12	Manual	Pass

This table summarizes all the test cases conducted through manual testing. Each test case is identified by a unique Test Case ID and corresponds to a specific Test Procedure ID. The test results show that all test items were successfully passed, indicating that the related functions met the expected requirements and no serious faults or abnormalities occurred during the testing process. This indicates that the core module of the current system performs well in terms of function implementation and has certain stability and availability.

5. Conclusion

The “Mobile Application for Exploring UKM” project successfully developed a centralized platform aimed at enhancing students’ access to essential campus information. Integrating features such as bus routes, the academic calendar, campus café locator, UKMFOLIO access, and news updates, the application streamlines information retrieval and reduce the need for multiple platforms. Furthermore, the use of Jetpack Compose delivers a modern, intuitive interface, while the integrated admin management system allows backend updates for news, events, and café listings, ensuring flexibility and long-term maintainability. Testing confirmed that the application meets its objectives in terms of functionality, usability, and reliability. However, certain constraints remain, including the absence of real-time bus and event updates, limited platform support, and the need for more personalized features. To address these, future work will focus on implementing live data integration, iOS compatibility,

personalized academic reminders, and in-app feedback mechanisms. Consequently, the application not only fulfills current student needs but also lays a strong foundation for continuous improvement. Overall, this project demonstrates the potential of a unified digital tool to enhance academic efficiency, promote campus engagement, and support the ongoing digital transformation of UKM.

6. APPRECIATION

Faculty of Technology and Information Science Grant Dr. Syaimak, LI YIHAO A197895.

7. REFERENCES

Smith, A., & Anderson, M. (2022). Technology and Student Engagement: How Mobile Technology Enhances Campus Life. Journal of Higher Education Technology, 45(3), 231-250.

Dube, L., & Loewen, P. (2015). "The Role of User Engagement in Mobile App Usage." International Journal of Mobile Human Computer Interaction, 7(4), 45-63.

Cyril, M. (2019). "Exploring Kotlin for Android Development: How the Language Enhances App Reliability and Developer Productivity." International Journal of Computer Applications, 178(5), 1-6.

Bishop, C. (2014). Mobile Apps for Higher Education: A Comprehensive Guide to Best Practices. Education Technology Publications.

UKM Educational system: https://ukmfolio.ukm.my/

UKM online learning platform: https://smplucee.ukm.my/smpwebbi

Ally, M., & Wark, N. (2020). "Using mobile learning to improve student engagement and academic performance." *Educational Technology Research and Development*, 68(4), 1255-1275.]

MyUKM Application

MyLboro Application

MIT Mobile Application

LI YIHAO (A197895)
Dr. Syaimak Abdul Shukor
Faculty of Information Science & Technology

Universiti Kebangsaan Malaysia
43600 Bangi, Selangor