

**CSE6224 Software Requirements Engineer TT3L**

**SRS**

**GROUP 1**

**Prepared by :**

|  |  |  |
| --- | --- | --- |
| **NAME** | **STUDENT ID** | **EMAIL** |
| MUHAMMAD NAQIB BIN ZULL AZRI | 1211112306 | [1211112306@student.mmu.edu.my](mailto:1211112306@student.mmu.edu.my) |
| MUHAMMAD HARITH AIMAN BIN MUHD ZULKAPLI | 1211112350 | [1211112350@student.mmu.edu.my](mailto:1211112350@student.mmu.edu.my) |
| DHARVIN DARAN A L ELANGOO | 1231303548 | [1231303548@student.mmu.edu.my](mailto:1231303548@student.mmu.edu.my) |
| GOH YENG XUN | 1231303430 | [1231303430@student.mmu.edu.my](mailto:1231303430@student.mmu.edu.my) |

Table of Contents

[1.0 Introduction 6](#_Toc201531705)

[1.1 Purpose 6](#_Toc201531706)

[1.2 Scope 6](#_Toc201531707)

[1.3 Product Overview 7](#_Toc201531708)

[1.3.1 Product Perspective 7](#_Toc201531709)

[1.3.2 Product Function 7](#_Toc201531710)

[1.3.3 User Characteristics 8](#_Toc201531711)

[1.3.4 Limitation 8](#_Toc201531712)

[1.4 Definition 9](#_Toc201531713)

[2.0 References 10](#_Toc201531714)

[3.0 Requirement 11](#_Toc201531715)

[3.1 Functions 11](#_Toc201531716)

[Figure 1.0 Use Case Diagram – Navigation System 11](#_Toc201531717)

[3.1.1 User Registration and Login 12](#_Toc201531718)

[3.1.2 View Campus Events 12](#_Toc201531719)

[3.1.3 Display Event Details 13](#_Toc201531720)

[3.1.4 Event Filtering 13](#_Toc201531721)

[3.1.5 Accessible Route Guidance 14](#_Toc201531722)

[3.1.6 No Events Message 14](#_Toc201531723)

[3.1.7 Class Schedule Integration 15](#_Toc201531724)

[3.1.8 Campus News Feed Integration 15](#_Toc201531725)

[3.1.9 Customizable Notification Filters 16](#_Toc201531726)

[3.1.10 Smart Suggestions / Context-Aware Alerts 16](#_Toc201531727)

[3.1.11 Notification Fatigue Prevention 17](#_Toc201531728)

[1st use case 18](#_Toc201531729)

[Figure 2.0 State Diagram – User Registration or Login 19](#_Toc201531730)

[2nd Use case 20](#_Toc201531731)

[Figure 3.0 State Diagram – User View Campus Event 21](#_Toc201531732)

[3rd Use case 22](#_Toc201531733)

[Figure 4.0 State Diagram – User Get Route 23](#_Toc201531734)

[4rd Use case 24](#_Toc201531735)

[Figure 5.0 State Diagram – User Real-time Update 25](#_Toc201531736)

[5th Use case 26](#_Toc201531737)

[Figure 6.0 State Diagram – Admin Update Event 27](#_Toc201531738)

[6th Use case 28](#_Toc201531739)

[Figure 7.0 State Diagram – Admin Update Facility 29](#_Toc201531740)

[7th Use case 30](#_Toc201531741)

[Figure 8.0 State diagram for class schedule integration 31](#_Toc201531742)

[8th Use case 32](#_Toc201531743)

[Figure 9.0 State diagram for campus news feed 33](#_Toc201531744)

[9th Use case 34](#_Toc201531745)

[Figure 10.0 State diagram for customizable notification filters 35](#_Toc201531746)

[10th Use case 36](#_Toc201531747)

[Figure 11.0 State diagram for smart suggestions and context-aware alerts 37](#_Toc201531748)

[11th Use case 38](#_Toc201531749)

[Figure 12.0 State diagram for notification fatigue prevention 39](#_Toc201531750)

[3.2 Performance Requirements 40](#_Toc201531751)

[3.3 Usability Requirement 42](#_Toc201531752)

[3.4 Interface Requirement 45](#_Toc201531753)

[3.4.1 User Interface 45](#_Toc201531754)

[3.4.2 Hardware Interface 49](#_Toc201531755)

[3.4.3 Software Interface 49](#_Toc201531756)

[3.4.4 Communications Interface 51](#_Toc201531757)

[3.5 Logical Database Requirement 52](#_Toc201531758)

[Data Entities and Attributes 52](#_Toc201531759)

[Table 3.5.1 Data Entities and Attributes 53](#_Toc201531760)

[Relationship Between Entities 54](#_Toc201531761)

[Entity Relationship Diagram 55](#_Toc201531762)

[3.6 Design Constraints 56](#_Toc201531763)

[3.6.1 Platform Compatibility 56](#_Toc201531764)

[3.6.2 Accessibility Standards 57](#_Toc201531765)

[3.6.3 Integration Requirements 58](#_Toc201531766)

[3.6.4 Technology Constraints 59](#_Toc201531767)

[3.6.5 Security and Privacy Compliance 59](#_Toc201531768)

[3.6.6 Network Dependency 60](#_Toc201531769)

[3.6.7 Maintainability 60](#_Toc201531770)

[3.6.8 Database Constraints 61](#_Toc201531771)

[3.6.9 Third-Party API Reliability (for Class Schedule & News Feed) 62](#_Toc201531772)

[3.6.10 User Notification Preferences Security 63](#_Toc201531773)

[3.6.11 Notification Delivery Platform Compliance 63](#_Toc201531774)

[3.6.12 Real-Time Suggestion Accuracy 64](#_Toc201531775)

[3.6.13 Notification Batching Logic 64](#_Toc201531776)

[3.7 Software System Attributes 65](#_Toc201531777)

[3.7.1 Availability 65](#_Toc201531778)

[3.7.2 Security 66](#_Toc201531779)

[3.7.3 Accessibility 66](#_Toc201531780)

[3.7.4 Reliability 67](#_Toc201531781)

[3.7.5 Maintainability 68](#_Toc201531782)

[3.7.6 Portability 69](#_Toc201531783)

[3.7.7 Scalability 70](#_Toc201531784)

[3.7.8 Auditability 70](#_Toc201531785)

[3.7.9 Performance 71](#_Toc201531786)

[3.7.10 Class Schedule Accuracy 71](#_Toc201531787)

[3.7.11 News Feed Timeliness 72](#_Toc201531788)

[3.7.12 Notification Filter Effectiveness 72](#_Toc201531789)

[3.7.13 Suggestion Relevance 73](#_Toc201531790)

[3.8 Supporting Information 74](#_Toc201531791)

[3.8.1 References and Background information that supports the development of MMUAccess 74](#_Toc201531792)

[3.8.2 Validation Sessions 75](#_Toc201531793)

[3.8.3 Defect Summary 76](#_Toc201531794)

[A. Content Defect 76](#_Toc201531795)

[B. Documentation Defect 78](#_Toc201531796)

[C. Agreement Defect 79](#_Toc201531797)

[3.8.4 Conflict Analysis 80](#_Toc201531798)

[3.8.5 Conflict Resolution 81](#_Toc201531799)

[3.8.6 Change Log 82](#_Toc201531800)

[3.8.7 Requirements Traceability Matrix 86](#_Toc201531801)

[3.8.8 Role in Requirements Validation, Negotiation & Management 87](#_Toc201531802)

[3.8.9 Version Control & Configuration Summary 87](#_Toc201531803)

[3.8.10 Supporting Document 88](#_Toc201531804)

[4.0 Verification 89](#_Toc201531805)

[4.1 Requirement Traceability 89](#_Toc201531806)

[4.2 Verification Techniques 89](#_Toc201531807)

[4.3 Stakeholder and Verification Summary 90](#_Toc201531808)

[4.3.1 Students - Verification Test Cases 90](#_Toc201531809)

[4.3.2 Admin (Staff) VerificationTest Cases 92](#_Toc201531810)

[4.3.3 Project Developer Verification Test Case 93](#_Toc201531811)

[5.0 Appendices 94](#_Toc201531812)

[5.1 Assumptions and Dependencies 94](#_Toc201531813)

[5.1.1 User Connectivity Prerequisite: 94](#_Toc201531814)

[5.1.2 MMU Backend System Integration and Data Provision: 94](#_Toc201531815)

[5.1.2 Administrative Data Maintenance and Accuracy: 94](#_Toc201531816)

[5.1.3 End-User Device and Browser Compatibility: 95](#_Toc201531817)

[5.2 Acronyms and Abbreviations 96](#_Toc201531818)

## 

# Introduction

## 1.1 Purpose

The purpose of this document is to specify the software requirements for MMUAccess, a comprehensive campus accessibility and information system. This system is designed to seamlessly assist all users (students, staff, and visitors) in navigating the Multimedia University campus, with a particular emphasis on providing inclusive experiences for users with diverse accessibility needs. The platform will leverage detailed building and facility information to provide real-time, accessible route guidance, account for event-related impacts on facilities and routes, and present a consolidated view of campus events and official news updates. Furthermore, MMUAccess aims to empower users through personalized class schedules, customizable preferences for notifications and suggestions, and secure device session management. The system also incorporates robust logging of administrative updates to maintain data integrity. Ultimately, MMUAccess seeks to significantly enhance the university's digital infrastructure by fostering inclusive campus navigation and ensuring all users have access to timely, relevant, and personalized campus information.

## 1.2 Scope

MMUAccess shall facilitate mainly the following operations

1. **User Management and Personalization:**
   1. Secure user authentication and session management for students, staff, and visitors, tracking device usage.
   2. Allow users to define and save detailed accessibility needs within their profiles.
   3. Provide comprehensive user preferences for notification types, urgency levels, delivery channels, batch settings, do-not-disturb periods, smart suggestions, and mute options.
2. **Accessible Campus Navigation:**
   1. Provide real-time, accessible route guidance based on user accessibility needs, utilizing comprehensive building and facility data.
   2. Incorporate real-time route adjustments based on various factors, including temporary event-related impacts on facilities or routes.
3. **Event and Campus Information Dissemination:**
   1. Display live updates and details on campus events.
   2. Integrate with MMU’s official event calendar to guide users to accessible locations for listed events.
   3. Provide official campus news, announcements, and updates through a dedicated news feed.
4. **Personalized Timetables:**
   1. Store and display personalized class schedules for students, including course details, venues, and lecturers.
5. **Notifications and Suggestions:**
   1. Log and manage all sent notifications, including delivery status and user responses, adhering to user preferences.
   2. Deliver personalized suggestions to users based on context and preferences, logging their content and status.
6. **System Administration and Auditing:**
   1. Maintain a detailed log of all administrative updates to core entities and data, linked to the responsible user.

## 1.3 Product Overview

### 1.3.1 Product Perspective

MMUAccess is a new, independent system designed specifically for Multimedia University. However, it will interact with existing university systems such as the campus facilities management database and the official event calendar system. The system is developed as a responsive web and mobile application to ensure accessibility across multiple platforms.

This product will serve as an enhancement to the university's infrastructure by supporting inclusive campus navigation. It provides real-time route adjustments and helps users with disabilities find the most suitable paths and access points.

### 1.3.2 Product Function

MMUAccess shall facilitate the following core functions:

1. **Generate Accessible Navigation Routes**: The system will provide navigation routes that avoid stairs, blocked paths, or other obstacles for users with accessibility needs.
2. **Event Viewing**: Users will be able to view detailed information about campus events, including their locations, times, and accessibility status.
3. **Get Route to Event**: After viewing an event, users can receive accessible route guidance to the event location.
4. **Real-time Updates**: The system will provide real-time notifications regarding any changes that may affect event locations or accessible paths.

### 1.3.3 User Characteristics

i. Students, staff, and visitors of MMU who require assistance navigating the campus, especially users with mobility, visual, or auditory impairments.  
ii. Users with varying levels of technical experience, from tech-savvy individuals to those with limited digital skills.  
iii. Administrative staff responsible for updating facility information and event accessibility details.  
iv. Users accessing the system from different devices, including smartphones, tablets, and desktop computers.

### 1.3.4 Limitation

i. The system depends on timely data updates from MMU staff for construction zones, elevator outages, and event information.  
ii. Internet connection is required for accessing real-time updates and using the application.  
iii. MMUAccess is limited to the MMU campus environment and will not provide navigation outside university grounds.  
iv. The system’s route accuracy may be affected by incomplete or outdated facility data.

## 

## 1.4 Definition

|  |  |
| --- | --- |
| **Term** | **Definition** |
| ***Accessible Route*** | A navigation route that avoids obstructions like stairs or blocked pathways in order to accommodate people with disabilities. |
| ***Admin*** | The system's event and facility information is updated by authorized university staff. |
| ***Accessibility Tags*** | Metadata that indicates if a venue or event satisfies accessibility requirements. |
| ***Event Calendar*** | MMU's official campus event scheduling system, which is connected to MMUAccess. |
| ***Facilities Database*** | Data about campus infrastructure, including ramps, lifts, and walkways, that is kept up to date by the university. |
| ***MMUAccess*** | The name of the proposed campus accessibility navigation system for MMU. |
| ***Real-time Updates*** | Instantaneous notification of changes, such construction, event relocation, or lift outages. |
| ***RESTful API*** | A web service API used for system integration that adheres to REST standards. |
| ***User*** | Any user of MMUAccess, including visitors, employees, and students. |
| ***WCAG*** | International guidelines for web accessibility are called Web Content Accessibility Guidelines. |

# 2.0 References

[1] IEEE Std 29148-2018, *Systems and software engineering — Life cycle processes — Requirements engineering*, IEEE Standards Association, 2018.

[2] N. Kano, N. Seraku, F. Takahashi, and S. Tsuji, “Attractive quality and must-be quality,” *Journal of the Japanese Society for Quality Control*, vol. 14, no. 2, pp. 39–48, 1984.

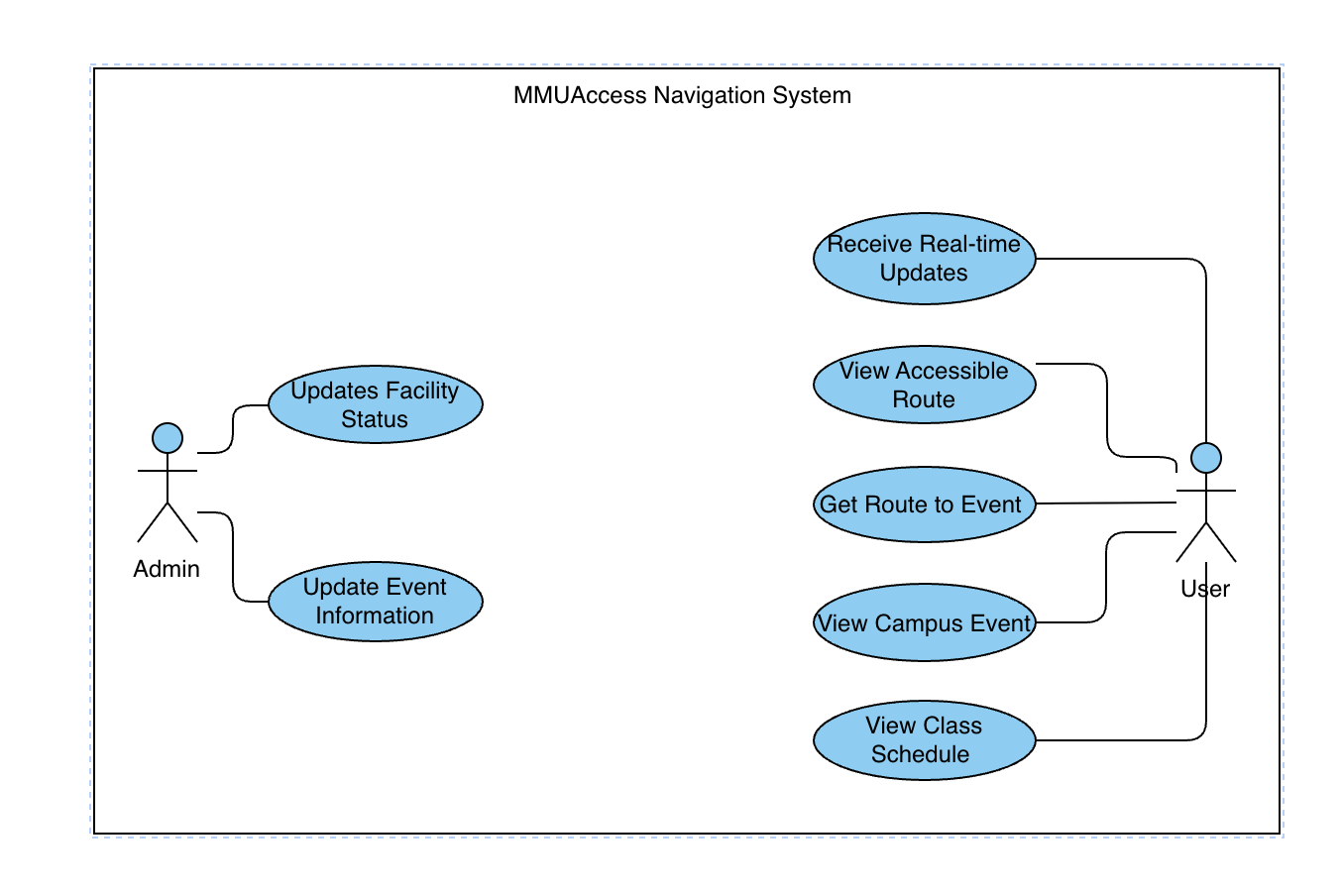
[6] Personal Data Protection Act (PDPA) 2010, Malaysia. [Online]. Available:<https://www.pdp.gov.my>. [Accessed: May 2025].

# 3.0 Requirement

## 3.1 Functions

This section details the functional requirements of the **MMUAccess Navigation System**, starting with the overall requirement, followed by specific requirements for each feature of the system.

Figure 2.0 shows the overall use case of MMUAccess, which supports **both Users and Admin** interactions with the system.



#### Figure 1.0 Use Case Diagram – Navigation System

#### 

### 3.1.1 User Registration and Login

|  |  |
| --- | --- |
| **REQ\_F001** | User Registration and Login |
| **Version** | 1.0 |
| **Description** | The system shall allow users to register for an account or log in using valid credentials. |
| **Priority** | High |
| **Acceptance Criteria** | User can successfully register or login; invalid credentials show error. |
| **Author** | Naqib |

### 3.1.2 View Campus Events

|  |  |
| --- | --- |
| **REQ\_F002** | View Campus Events |
| **Version** | 1.0 |
| **Description** | The system shall allow users to access and view a list of current campus events. |
| **Priority** | High |
| **Acceptance Criteria** | User can see a list of events, including event name, date, time, and location. |
| **Author** | Naqib |

### 3.1.3 Display Event Details

|  |  |
| --- | --- |
| **REQ\_F003** | Display Event Details |
| **Version** | 1.0 |
| **Description** | The system shall display event name, location, time, and accessibility information |
| **Priority** | Medium |
| **Acceptance Criteria** | When user clicks on an event, detailed information is shown |
| **Author** | Naqib |

### 3.1.4 Event Filtering

|  |  |
| --- | --- |
| **REQ\_F004** | Event Filtering |
| **Version** | 1.0 |
| **Description** | The system shall allow users to filter events based on date and category |
| **Priority** | Medium |
| **Acceptance Criteria** | User can filter events and view only relevant results |
| **Author** | Naqib |

### 3.1.5 Accessible Route Guidance

|  |  |
| --- | --- |
| **REQ\_F005** | Accessible Route Guidance |
| **Version** | 1.0 |
| **Description** | The system shall allow users to get accessible route guidance to event locations |
| **Priority** | High |
| **Acceptance Criteria** | When user selects "Get Route", an accessible path is shown from user's current location to the event. |
| **Author** | Naqib |

### 3.1.6 No Events Message

|  |  |
| --- | --- |
| **REQ\_F006** | No Events Message |
| **Version** | 1.0 |
| **Description** | The system shall display a message if no upcoming events are available |
| **Priority** | Low |
| **Acceptance Criteria** | If the event list is empty, a clear message is shown to the user |
| **Author** | Naqib |

### 3.1.7 Class Schedule Integration

|  |  |
| --- | --- |
| **REQ\_F007** | Class Schedule Integration |
| **Version** | 1.1 |
| **Description** | The system shall allow student users to view and sync their personalized class schedule, including course name, time, venue, and updates from the university timetable |
| **Priority** | High |
| **Acceptance Criteria** | Students can access a complete list of upcoming classes with accurate schedule data; changes in timetable are reflected in the app within 2 minutes |
| **Author** | Group 2 |

### 3.1.8 Campus News Feed Integration

|  |  |
| --- | --- |
| **REQ\_F008** | Campus News Feed Integration |
| **Version** | 1.1 |
| **Description** | The system shall display a campus news feed on the home page, providing official announcements, campus updates, and policy news relevant to users. |
| **Priority** | Medium |
| **Acceptance Criteria** | Users can view a list of current news headlines and summaries; tapping a headline displays the full article |
| **Author** | Group 2 |

### 3.1.9 Customizable Notification Filters

|  |  |
| --- | --- |
| **REQ\_F009** | Customizable Notification Filters |
| **Version** | 1.1 |
| **Description** | The system shall allow users to customize notification preferences, including filtering by type (event, facility, news), urgency, and delivery channel |
| **Priority** | High |
| **Acceptance Criteria** | Users can set notification preferences in the app settings and only receive notifications according to their choices |
| **Author** | Group 2 |

### 3.1.10 Smart Suggestions / Context-Aware Alerts

|  |  |
| --- | --- |
| **REQ\_F010** | Smart Suggestions / Context-Aware Alerts |
| **Version** | 1.1 |
| **Description** | The system shall provide users with personalized, context-aware suggestions or reminders (e.g., “You have 15 minutes before your next class”) based on timetable, location, or current campus status |
| **Priority** | Medium |
| **Acceptance Criteria** | Users receive timely smart suggestions relevant to their class schedule and location; suggestions are delivered at least 10 minutes before the related event when possible |
| **Author** | Group 2 |

### 3.1.11 Notification Fatigue Prevention

|  |  |
| --- | --- |
| **REQ\_F011** | Notification Fatigue Prevention |
| **Version** | 1.1 |
| **Description** | The system shall provide notification fatigue prevention features, such as batching non-urgent notifications and allowing users to mute or snooze repeated alerts. |
| **Priority** | Medium |
| **Acceptance Criteria** | Users can enable notification batching or “Do Not Disturb” mode; the system limits non-urgent notifications to no more than 3 per hour by default |
| **Author** | Group 2 |

### 1st use case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | UC001 | | **Version** | 1.0 |
| **Feature** | F001 User Registration/Login | | | |
| **Purpose** | To allow new users to register or existing users to log in to access MMUAccess. | | | |
| **Actor** | User | | | |
| **Trigger** | User opens the app and selects either “Register” or “Login”. | | | |
| **Precondition** | App is launched on a supported device (web/mobile) | | | |
| **Scenario Name** | **Step** | **Action** | | |
| **Main Flow** | 1 | User selects “Login” or “Register” from the app’s welcome screen | | |
| 2 | The system displays username/email and password input fields. | | |
| 3 | User enters credentials or registration details (name, ID, email, password). | | |
| 4 | System validates input and checks the database | | |
| 5 | If valid, system logs in or registers the user and redirects to main menu. | | |
| **Alternative Flow – Invalid Credential** | 3.1 | User inputs incorrect or empty credentials. | | |
| 3.2 | System displays an error: “Invalid username or password.” | | |
| 3.3 | Return to step 2 | | |
| **Alternative Flow – Already Registered Email** | 3.1 | User enters an email that already exists during registration. | | |
| 3.2 | System displays error: “Email already registered. Try logging in instead.” | | |
| 3.3 | Return to step 2 | | |
| **Rules** | Password must be at least 8 characters | | | |
| Email must be in valid format | | | |
| **Author** | Naqib | | | |

A diagram of a computer program

AI-generated content may be incorrect.

#### Figure 2.0 State Diagram – User Registration or Login

### 2nd Use case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | UC002 | | **Version** | 1.0 |
| **Feature** | F002 View Campus Event | | | |
| **Purpose** | To allow users to view a list of upcoming campus events and their details. | | | |
| **Actor** | User | | | |
| **Trigger** | User selects “Events” from the app’s navigation menu. | | | |
| **Precondition** | User has opened the app (login) | | | |
| **Scenario Name** | **Step** | **Action** | | |
| **Main Flow** | 1 | User selects “Events” from the menu. | | |
| 2 | System retrieves event data from the university's event calendar database. | | |
| 3 | System displays a list of events with event name, date, time, and venue info. | | |
| 4 | User selects an event to view full details. | | |
| **Alternative Flow – No Events Available** | 1.1 | System finds no upcoming or active events in the database. | | |
| 1.2 | System displays message: “No upcoming events at the moment.” | | |
| 1.3 | System returns to home screen or keeps event page open for refresh. | | |
| **Rules** | Events must contain a title, date/time, venue, and accessibility tags (if applicable). | | | |
| Events are sorted by date, showing the soonest events first. | | | |
| **Author** | Naqib | | | |

A diagram of a event

AI-generated content may be incorrect.

#### Figure 3.0 State Diagram – User View Campus Event

### 3rd Use case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | UC003 | | **Version** | 1.0 |
| **Feature** | F003 Get Route to Event | | | |
| **Purpose** | To allow users to view the accessible route to a selected campus event. | | | |
| **Actor** | User | | | |
| **Trigger** | User selects an event and clicks “Get Route” button. | | | |
| **Precondition** | Events have valid location info; users are connected to the internet. | | | |
| **Scenario Name** | **Step** | **Action** | | |
| **Main Flow** | 1 | User selects an event from the list. | | |
| 2 | User clicks the “Get Route” or navigation icon. | | |
| 3 | System requests user’s current location (with permission). | | |
| 4 | System calculates and displays the most accessible route to the event venue. | | |
| **Alternative Flow – Location Access Denied** | 3.1 | System finds no upcoming or active events in the database. | | |
| 3.2 | System displays message: “No upcoming events at the moment.” | | |
| 3.3 | System returns to home screen or keeps event page open for refresh. | | |
| **Alternative Flow – No Accessible Route Found** | 4.1 | System cannot find an accessible route due to outages or restrictions. | | |
| 4.2 | System displays message: “No accessible route currently available.” | | |
| 4.3 | Optionally provides alternate navigation options or returns to event view. | | |
| **Rules** | Routes must consider user accessibility preferences | | | |
| Event venue must be within campus boundaries. | | | |
| **Author** | Naqib | | | |

A diagram of a route

AI-generated content may be incorrect.

#### Figure 4.0 State Diagram – User Get Route

### 

### 4rd Use case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | UC004 | | **Version** | 1.0 |
| **Feature** | F004 Receive Real-time Updates | | | |
| **Purpose** | To notify users of live changes affecting campus accessibility and events. | | | |
| **Actor** | User | | | |
| **Trigger** | System detects an update in facilities or event changes from admin input. | | | |
| **Precondition** | User has an active internet connection and notification permissions enabled. | | | |
| **Scenario Name** | **Step** | **Action** | | |
| **Main Flow** | 1 | System checks for updates from facility and event databases. | | |
| 2 | If updates are found, system verifies if they affect user’s saved routes/events. | | |
| 3 | System sends a real-time notification to the user. | | |
| 4 | User opens the app and views the affected route or event change. | | |
| **Alternative Flow – Notifications Disabled** | 3.1 | User has disabled notifications. | | |
| 3.2 | System logs the update silently and displays a badge icon in the app. | | |
| 3.3 | User sees alert manually when navigating or opening the relevant section. | | |
| **Rules** | Updates include elevator outages, blocked paths, relocated events, or urgent campus notices. | | | |
| **Author** | Naqib | | | |

A diagram of a system

AI-generated content may be incorrect.

#### Figure 5.0 State Diagram – User Real-time Update

### 5th Use case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | UC005 | | **Version** | 1.0 |
| **Feature** | F005 Update Event Information | | | |
| **Purpose** | To allow admin users to add, edit, or delete campus event information. | | | |
| **Actor** | Admin | | | |
| **Trigger** | Admin logs into the system and selects the “Manage Events” option. | | | |
| **Precondition** | Admin is authenticated and authorized to access the admin panel. | | | |
| **Scenario Name** | **Step** | **Action** | | |
| **Main Flow** | 1 | Admin logs in to the MMUAccess system. | | |
| 2 | Admin selects “Manage Events” from the dashboard. | | |
| 3 | Admin adds a new event or selects an existing event to edit or delete. | | |
| 4 | System updates the event database accordingly. | | |
| 5 | System confirms success and reflects changes in the user interface. | | |
| **Alternative Flow – Missing Required Fields** | 3.1 | Admin submits an event form with missing required information. | | |
| 3.2 | System displays an error message prompting admin to complete all fields. | | |
| 3.3 | Return to event form | | |
| **Rules** | Event entries must include title, date, time, venue, and optional accessibility tags. | | | |
| Edits must be logged with timestamps and admin ID. | | | |
| Deleted events must be archived and not immediately removed from the database. | | | |
| **Author** | Naqib | | | |

A diagram of a function

AI-generated content may be incorrect.

#### Figure 6.0 State Diagram – Admin Update Event

### 6th Use case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | UC006 | | **Version** | 1.0 |
| **Feature** | F006 Update Facility Status | | | |
| **Purpose** | To allow admin users to update the status of campus facilities (e.g., elevators, paths). | | | |
| **Actor** | Admin | | | |
| **Trigger** | Admin logs in and selects “Update Facility” from the admin dashboard. | | | |
| **Precondition** | Admin is authenticated and authorized to access facility management. | | | |
| **Scenario Name** | **Step** | **Action** | | |
| **Main Flow** | 1 | Admin logs into the MMUAccess admin panel | | |
| 2 | Admin selects “Update Facility” from the dashboard | | |
| 3 | Admin selects a facility (e.g., elevator, ramp) to mark as active/inactive | | |
| 4 | System saves the update in the facility database | | |
| 5 | System reflects the update on the user interface in real time | | |
| **Alternative Flow – Missing Required Fields** | 3.1 | Admin attempts to update a facility not registered in the system | | |
| 3.2 | System shows error: “Facility not found. Please register or contact support.” | | |
| 3.3 | Admin returns to facility selection | | |
| **Rules** | Facility entries must include type, location, and status. | | | |
| Updates should be timestamped and linked to admin user ID. | | | |
| Changes should trigger alerts for users currently affected (e.g., on a route using that facility). | | | |
| **Author** | Naqib | | | |

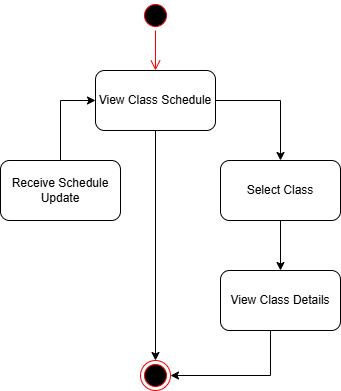
A diagram of a facility

AI-generated content may be incorrect.

#### Figure 7.0 State Diagram – Admin Update Facility

### 7th Use case

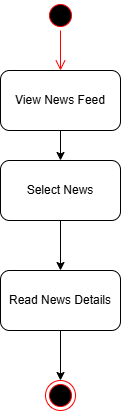
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | **UC007** | **Version** | 1.1 |
| **Feature** | F007 Class Schedule Integration | | |
| **Purpose** | To allow students to view their personalized class timetable, including details and real-time updates. | | |
| **Trigger** | Student selects "Class Schedule" from the navigation menu. | | |
| **Author** | Group 2 | | |
| **Actor** | Student | | |
| **Precondition** | Student is logged in and has an assigned class schedule. | | |
| **Scenario Name** | **Step** | **Action** | |
| **Main Flow** | 1 | Student opens class schedule page. | |
| 2 | System retrieves and displays current and upcoming classes. | |
| 3 | Student clicks a class for details/location. | |
| **Alternative Flow – No Classes** | 1.1 | No schedule found for this student. | |
| 1.2 | System displays: "No classes scheduled." | |
| **Rules** | Schedule includes course name, time, location, lecturer. | | |



#### Figure 8.0 State diagram for class schedule integration

### 8th Use case

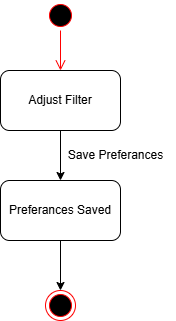
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | **UC008** | **Version** | **1.1** |
| **Feature** | F008 Campus News Feed Integration | | |
| **Purpose** | To allow users to view campus news, announcements, and alerts from a centralized feed. | | |
| **Trigger** | User selects "News Feed" or opens the app homepage. | | |
| **Author** | Group 2 | | |
| **Actor** | User | | |
| **Precondition** | News feed data is available from the university backend. | | |
| **Scenario Name** | **Step** | **Action** | |
| **Main Flow** | 1 | User opens the news feed. | |
| 2 | System displays latest news headlines and summaries. | |
| 3 | User selects a headline to read full details. | |
| **Alternative Flow – No News** | 2.1 | No news found. | |
| 2.2 | System displays: "No news available." | |
| **Rules** | News items include title, date, summary, and details. | | |



#### Figure 9.0 State diagram for campus news feed

### 9th Use case

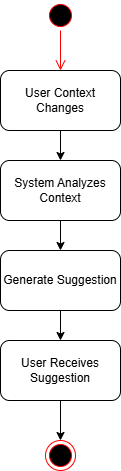
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | **UC009** | **Version** | **1.1** |
| **Feature** | F009 Customizable Notification Filters | | |
| **Purpose** | To allow users to set and manage their notification preferences, filtering by type, urgency, and channel. | | |
| **Trigger** | User navigates to "Notification Settings". | | |
| **Author** | Group 2 | | |
| **Actor** | User | | |
| **Precondition** | User is logged in. | | |
| **Scenario Name** | **Step** | **Action** | |
| **Main Flow** | 1 | User opens notification settings page. | |
|  | 2 | User selects desired notification types/urgency/channels. | |
|  | 3 | System saves and applies these preferences. | |
| **Alternative Flow – No Preferences Set** | 2.1 | User does not select any notification type. | |
|  | 2.2 | System prompts: "Please select at least one notification type." | |
| **Rules** | Preferences are saved per user account. | | |



#### Figure 10.0 State diagram for customizable notification filters

### 10th Use case

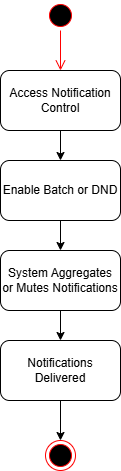
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | **UC010** | **Version** | **1.0** |
| **Feature** | F010 Smart Suggestions / Context-Aware Alerts | | |
| **Purpose** | To proactively provide users with relevant reminders or route suggestions based on schedule, location, or event data. | | |
| **Trigger** | System detects relevant upcoming class/event or context change. | | |
| **Author** | Group 2 | | |
| **Actor** | User | | |
| **Precondition** | User's schedule/location/context is available. | | |
| **Scenario Name** | **Step** | **Action** | |
| **Main Flow** | 1 | System checks user schedule, location, and event data. | |
| 2 | If relevant, system sends personalized suggestion or alert. | |
| **Alternative Flow – No Relevant Context** | 2.1 | No upcoming events or actionable context. | |
| 2.2 | No suggestion or alert is generated. | |
| **Rules** | Suggestions must be timely and context-specific. | | |



#### Figure 11.0 State diagram for smart suggestions and context-aware alerts

### 11th Use case

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | **UC011** | **Version** | **1.0** |
| **Feature** | F011 Notification Fatigue Prevention | | |
| **Purpose** | To help users reduce or mute non-urgent notifications, batch alerts, and enable “Do Not Disturb” periods. | | |
| **Trigger** | User accesses notification controls/settings. | | |
| **Author** | Group 2 | | |
| **Actor** | User | | |
| **Precondition** | User is logged in. | | |
| **Scenario Name** | **Step** | **Action** | |
| **Main Flow** | 1 | User opens notification control or DND settings. | |
|  | 2 | User sets notification batching or Do Not Disturb period. | |
|  | 3 | System applies new notification rules. | |
| **Alternative Flow – All Notifications Disabled** | 2.1 | User disables all notifications. | |
|  | 2.2 | System displays: "Notifications are currently disabled." | |
| **Rules** | System should not send more than 3 non-urgent notifications per hour by default. | | |



#### Figure 12.0 State diagram for notification fatigue prevention

## 

## 3.2 Performance Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Version** | **Description** | **Priority** | **Acceptance Criteria** | **Author / Group** |
| REQ\_P001: Response Time | 1.0 | The system shall display event and route search results within 3 seconds of user request under normal load. | High | 95% of user requests are completed within 3 seconds during standard operation. | Naqib |
| REQ\_P002: Concurrent Users | 1.0 | The system shall support at least 500 concurrent users (web and mobile) without noticeable performance degradation. | High | System performance (response time, error rate) remains within acceptable limits with 500 active sessions. | Naqib |
| REQ\_P003: System Availability | 1.0 | The system shall maintain an uptime of at least 99.5% during university operating hours, excluding scheduled maintenance. | High | Monthly uptime report shows ≥99.5% availability. | Group 2 |
| REQ\_P004: Data Backup & Recovery | 1.0 | The system shall perform automatic daily backups of all critical data and be able to recover within 10 minutes after an unexpected outage. | Medium | Backup and restore processes are tested monthly; recovery time ≤10 minutes. | Group 2 |
| REQ\_P005: Scalability | 1.0 | The system shall be able to scale up to 1000 concurrent users within 6 months after initial deployment with minimal downtime. | Medium | System can be upgraded (e.g., via cloud resources) to meet future growth. | Group 2 |
| REQ\_P006: Class Schedule Sync Latency | 1.1 | The system shall update a student's class schedule in the app within 2 minutes of any change in the university's timetable system. | High | 95% of schedule changes are reflected within 2 minutes of update in source system. | Group 2 |
| REQ\_P007: News Feed Load Time | 1.1 | The system shall display campus news headlines within 2 seconds of the user opening the news feed, under normal network conditions. | Medium | 95% of news feed openings display headlines within 2 seconds. | Group 2 |
| REQ\_P008: Notification Filter Application Time | 1.1 | Changes to notification filter preferences shall take effect within 5 seconds after user confirmation. | High | 95% of filter changes are active and reflected in notification delivery within 5 seconds. | Group 2 |
| REQ\_P009: Smart Suggestions Delivery Time | 1.1 | Context-aware suggestions (e.g., class reminders) shall be delivered to the user at least 10 minutes before the relevant event when possible. | Medium | 95% of suggestions are delivered on time based on user context and schedule. | Group 2 |
| REQ\_P010: Notification Fatigue Prevention Efficiency | 1.1 | The system shall limit non-urgent notification delivery to no more than 3 per hour by default for each user. | Medium | No user receives more than 3 non-urgent notifications per hour under default settings. | Group 2 |

## 

## 3.3 Usability Requirement

MMUAccess shall be designed with a strong focus on ease of use and accessibility for all users, including individuals with disabilities.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Version** | **Description** | **Priority** | **Acceptance Criteria** | **Author / Group** |
| REQ\_U001: Intuitive Navigation | 1.0 | The system shall provide a user interface with intuitive navigation and minimal learning curve, allowing users to complete core tasks (e.g., searching for accessible routes, viewing events) without requiring a manual. | High | At least 90% of users can complete key tasks (route search, event view) on their first attempt during usability testing, without external assistance. | Group1 |
| REQ\_U002: Task Efficiency | 1.0 | Users shall be able to complete core tasks (such as searching for accessible routes or viewing event locations) with no more than 3 clicks or taps. | High | Usability tests show that 95% of tasks are completed within 3 clicks/taps. | Group1 |
| REQ\_U003: Multi-Language Support | 1.0 | The system shall offer multi-language support, with English and Bahasa Malaysia as default options. | Medium | All user-facing screens and messages are available in both languages, and language can be switched at any time. | Group1 |
| REQ\_U004: User Guidance and Help | 1.0 | The system shall provide tooltips, onboarding guidance, and a help section to assist new users in using core features. | Medium | Usability testing confirms that users can access and understand help content, and at least 80% rate onboarding as helpful. | Group1 |
| REQ\_U005: Accessibility Features | 1.0 | The system shall include a high-contrast visual mode, adjustable font sizes, screen reader compatibility, and keyboard navigation for users with disabilities. | High | System passes accessibility tests based on WCAG 2.1 Level AA; features function as intended in assistive technology environments. | Group1 |
| REQ\_U006: Usability Compliance | 1.0 | System usability shall comply with ISO 9241 and WCAG 2.1 standards, aiming for at least Level AA conformance. | High | External usability and accessibility audit reports confirm compliance with relevant standards. | Group1 |
| REQ\_U007: User Satisfaction | 1.0 | The system shall achieve a System Usability Scale (SUS) score of at least 80 during acceptance testing. | Medium | Post-test SUS surveys with sample users average 80 or above. | Group2 |
| REQ\_U008: Class Schedule Usability | 1.0 | The system shall present class schedule information in a clear, calendar-style interface that allows students to access class details and navigation in no more than 2 taps. | High | Usability testing confirms 90% of students can access and interpret their schedule on first attempt, and use “navigate to class” without confusion. | Group 2 |
| REQ\_U009: News Feed Usability | 1.0 | The system shall display campus news and announcements in an easily scannable, prioritized list with headlines, summaries, and a one-tap option to read more. | Medium | 90% of users report they can easily find and read important news items during usability tests. | Group 2 |
| REQ\_U010: Notification Filter Usability | 1.0 | The system shall provide an intuitive notification settings page, enabling users to set and adjust notification filters with no more than 3 clicks or taps. | High | Usability tests confirm 90% of users can set preferences on first try, and only receive desired notifications. | Group 2 |
| REQ\_U011: Smart Suggestions Usability | 1.0 | Smart suggestions and alerts shall appear in a dedicated, non-intrusive notification area and use concise, context-relevant language. | Medium | 90% of test users report that suggestions are easy to notice and understand, and do not interfere with normal app usage. | Group 2 |
| REQ\_U012: Notification Fatigue Prevention Usability | 1.0 | The system shall make notification fatigue prevention settings (e.g., mute/batch) discoverable from the notification center or settings, and provide clear feedback when enabled. | Medium | 90% of users can find and enable notification batching or mute, and confirm their status, during usability testing. | Group 2 |

## 

## 3.4 Interface Requirement

### 3.4.1 User Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Version** | **Description** | **Priority** | **Acceptance Criteria** | **Author / Group** |
| REQ\_I001: User Interface – Responsiveness | 1.0 | The system shall provide a responsive web interface compatible with all major browsers (Chrome, Firefox, Safari, Edge) and mobile platforms, ensuring a seamless experience across devices. | High | The user interface displays correctly and functions as intended on at least 95% of tested devices and browsers. | Group 1 |
| REQ\_I002: User Interface – Mobile Optimization | 1.0 | The mobile interface shall be optimized for iOS and Android devices, supporting a range of screen sizes, orientations, and resolutions for consistent usability. | High | Usability testing confirms smooth operation and proper layout rendering on the latest versions of iOS and Android devices. | Group 1 |
| REQ\_I003: User Interface – Accessibility | 1.0 | All UI components (e.g., buttons, input fields, modals, navigation menus) shall conform to WCAG 2.1 Level AA standards. The design shall also follow the consistent patterns defined in the project’s [Design System/Style Guide]. | High | Accessibility audit reports indicate no critical issues. Compliance with WCAG 2.1 Level AA is confirmed through expert review and assistive technology testing. | Group 1 |
| REQ\_I004: Home Page Content Display | 1.0 | The home page shall prominently display key content including navigation shortcuts, a summary of upcoming events, and critical campus alerts in an accessible and prioritized layout. | Medium | Usability testing verifies users can easily identify and interact with navigation elements, review current/recent events, and access campus alerts effectively. | Group 1 |
| REQ\_I005: User Interface – Multimodal Interaction | 1.0 | Users shall be able to interact with the system using touch, keyboard navigation, and voice-based input methods. The UI must support compatibility with OS-level assistive technologies (e.g., iOS Voice Control, Android Voice Access, TalkBack, VoiceOver). | Medium | Usability tests confirm that key user tasks are successfully completed using touch, keyboard, and voice-based inputs across supported platforms. | Group 1 |
| REQ\_I012: User Interface – Class Schedule Display | 1.0 | The system shall display each user’s class schedule in a clear, calendar or timetable format, with color coding and one-tap access to class details and navigation. | High | Usability testing confirms that 90% of students can view and understand their class schedule at a glance, and access class detail/navigation with one tap. | Group 2 |
| REQ\_I013: User Interface – Campus News Feed | 1.0 | The home page shall include a scrollable campus news feed showing headlines, summaries, and tap-to-expand full news items. | Medium | Users can access the news feed from the home page and view full articles with one tap; layout is confirmed accessible and readable on web and mobile. | Group 2 |
| REQ\_I014: User Interface – Notification Settings | 1.0 | The system shall provide a notification settings page where users can filter notification types (events, news, facilities), urgency, and preferred channels via toggles or checkboxes. | High | Usability tests show that 90% of users can successfully set or update notification preferences without external help. | Group 2 |
| REQ\_I015: User Interface – Smart Suggestions / Context-Aware Alerts | 1.0 | Smart suggestions and alerts shall be delivered through a dedicated, unobtrusive notification panel within the app, using concise and context-relevant text. | Medium | Users report in usability testing that suggestions are clear, helpful, and do not interfere with regular app use. | Group 2 |
| REQ\_I016: User Interface – Notification Fatigue Controls | 1.0 | The system shall allow users to access batching, mute, and “Do Not Disturb” notification controls from the notification center or settings, with clear status indicators. | Medium | 90% of users can enable or disable batching/mute/DND in under three taps, and see active status indicators. | Group 2 |

### 

### 3.4.2 Hardware Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Version** | **Description** | **Priority** | **Acceptance Criteria** | **Author / Group** |
| REQ\_I006: Hardware Interface | 1.0 | The system shall operate on standard smartphones, tablets, and desktop computers without requiring special hardware beyond typical accessibility tools (e.g., screen readers). | Medium | System passes compatibility testing on a range of modern consumer devices. | Group 1 |

### 3.4.3 Software Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Version** | **Description** | **Priority** | **Acceptance Criteria** | **Author / Group** |
| REQ\_I007: Software Interface – Event Calendar Integration | 1.0 | The system shall integrate with the MMU Event Calendar API to retrieve event details and locations. | High | System successfully syncs event data from the API; integration is verified through functional testing. | Group 1 |
| REQ\_I008: Software Interface – Facilities Management Integration | 1.0 | The system shall integrate with the MMU Facilities Management Database for real-time campus infrastructure data. | High | System successfully retrieves and updates facility data; integration is verified through functional testing. | Group 1 |
| REQ\_I009: Software Interface – Secure API Standards | 1.0 | All external system integrations shall use secure RESTful APIs with JSON format for data exchange. | High | API endpoints use HTTPS and conform to security and data format standards; verified through code and penetration testing. | Group 1 |
| REQ\_I017: Software Interface – Class Schedule and News API Integration | 1.0 | The system shall integrate with the MMU Class Schedule API and the official Campus News Feed API to retrieve up-to-date user timetables and campus announcements. | High | System successfully syncs class schedule and news data from respective APIs; integration is verified through functional and real-time update tests. | Group 2 |
| REQ\_I018: Software Interface – Notification Delivery Platform | 1.0 | The system shall connect to an authorized third-party notification delivery platform to support push notifications and batch messaging in accordance with user preferences. | Medium | The platform delivers notifications reliably during integration testing and respects user filter/batching settings. | Group 2 |

### 

### 3.4.4 Communications Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Version** | **Description** | **Priority** | **Acceptance Criteria** | **Author / Group** |
| REQ\_I010: Communications Interface – Secure Data Transmission | 1.0 | All communications between client and server shall use HTTPS for secure data transmission. | High | All network traffic is encrypted; verified by network analysis and security audit. | Group 1 |
| REQ\_I011: Communications Interface – Cloud Hosting & Real-Time Updates | 1.0 | The system shall be hosted on a cloud platform supporting real-time updates and high availability. | High | Cloud deployment is operational and meets availability requirements; real-time update features function in system tests. | Group 1 |
| REQ\_I019: Communications Interface – Real-Time Notifications | 1.0 | The system shall support secure, real-time communication of personalized alerts, suggestions, and news using HTTPS and/or WebSocket or standard push notification services. | High | Alerts and news are delivered to users within 10 seconds under normal conditions, verified by test scenarios. | Group 2 |
| REQ\_I020: Communications Interface – User Notification Control | 1.0 | The system shall enable users to manage notification receipt (mute, batch, do not disturb) through secure communication between the client app and server. | Medium | User commands for notification control are securely transmitted and take effect within 5 seconds. | Group 2 |

## 3.5 Logical Database Requirement

This section defines the logical data requirements and relationships necessary to support the functionality of the MMUAccess platform. It outlines key data entities, their attributes, and the relationships between them to ensure seamless, accessible navigation and real-time event guidance.

### Data Entities and Attributes

|  |  |  |
| --- | --- | --- |
| **Entity Name** | **Description** | **Key Attributes** |
| **User** | Represents system users (students, staff, visitors) | UserID (PK), Name, Role, Email (Unique), PasswordHash, AccessibilityNeeds |
| **DeviceSession** | Stores session data. | SessionID (PK), UserID (FK), DeviceType, LoginTimestamp, LastActivityTimestamp |
| **Building** | Campus buildings. | BuildingID (PK), Name, Latitude, Longitude, Description |
| **Facility** | Accessibility facilities. | FacilityID (PK), BuildingID (FK), Type, Status, LastUpdated |
| **NavigationRoute** | Routes across campus. | RouteID (PK), StartPoint, EndPoint, IsAccessible, PathDetails, EstimatedTime |
| **Event** | Campus events. | EventID (PK), Title, Description, StartTime, EndTime, Location, IsAccessible, OrganizerContact |
| **EventImpact** | Impact of events on facilities/routes. | ImpactID (PK), EventID (FK), FacilityID (FK), RouteID (FK, optional), ImpactDescription |
| **UpdateLog** | Tracks admin updates. | UpdateID (PK), UserID (FK), EntityModified, ModificationDetails, Timestamp |
| **ClassSchedule** | Stores personalized class timetable for students | ScheduleID (PK), UserID (FK), CourseCode, CourseName, StartTime, EndTime, Venue, Lecturer, LastUpdated |
| **NewsFeed** | Stores official campus news, announcements, and updates | NewsID (PK), Title, Summary, Content, PublishDate, Category, Author, SourceURL |
| **UserPreferences** | Stores user-specific notification and filter settings | PrefID (PK), UserID (FK), NotificationTypes, UrgencyLevel, DeliveryChannel, BatchSetting, DNDPeriods, SmartSuggestionEnabled, MuteTypes |
| **NotificationLog** | Records all notifications sent, delivery status, and user response | NotificationID (PK), UserID (FK), Type, TimeSent, Batched (bool), Status (delivered/read/muted) |
| **SuggestionLog** | Logs personalized suggestions delivered to users | SuggestionID (PK), UserID (FK), Content, Timestamp, TriggerContext, Status |
| **Admin** | Represents administrators managing the system | AdminID (PK), Name, Email (Unique), PasswordHash, Role |

#### Table 3.5.1 Data Entities and Attributes

**Core Entities:**

* **Users:** Represents all system users, including students, staff, and administrators, along with their distinct roles and authentication details.
* **Campus Locations/Facilities:** Details various points of interest on campus, such as buildings, lecture halls, labs, and other amenities. This includes attributes like accessibility features and current status.
* **Events:** Captures information about campus events, encompassing details like time, location, description, and associated accessibility tags.
* **Navigation Routes:** Defines pathways between campus locations, with potential attributes for specific accessible routes (e.g., indicating presence of ramps or elevators).
* **Notifications:** Manages the creation and delivery of system alerts and personalized messages to users based on their preferences or relevant events.
* **Schedules :** Represents classes, linking users to specific times and locations for academic or personal planning.

**Key Relationships and System Support:**

The relationships between these entities are crucial for enabling the system's core functions and ensuring data integrity:

* **Events** are linked to specific **Campus Locations**.
* **Users** are associated with their individual **Schedules** and **Notification** preferences.
* **Navigation Routes** are established between various **Campus Locations**.

This high-level data model forms the foundational structure for storing and managing all essential information required for the MMUAccess system's operation.

### Relationship Between Entities

1. **One-to-Many**:

* One User can have multiple DeviceSession records.
* One Building can have multiple Facility records.
* One Event can impact multiple Facilities or NavigationRoutes.
* One User (admin) can create multiple UpdateLog entries.
* One User can have multiple ClassSchedule records
* One User can have many NotificationLog and SuggestionLog records.
* One Admin can manage multiple UpdateLog entries.

1. **One-to-One**:

* One User has one UserPreferences record

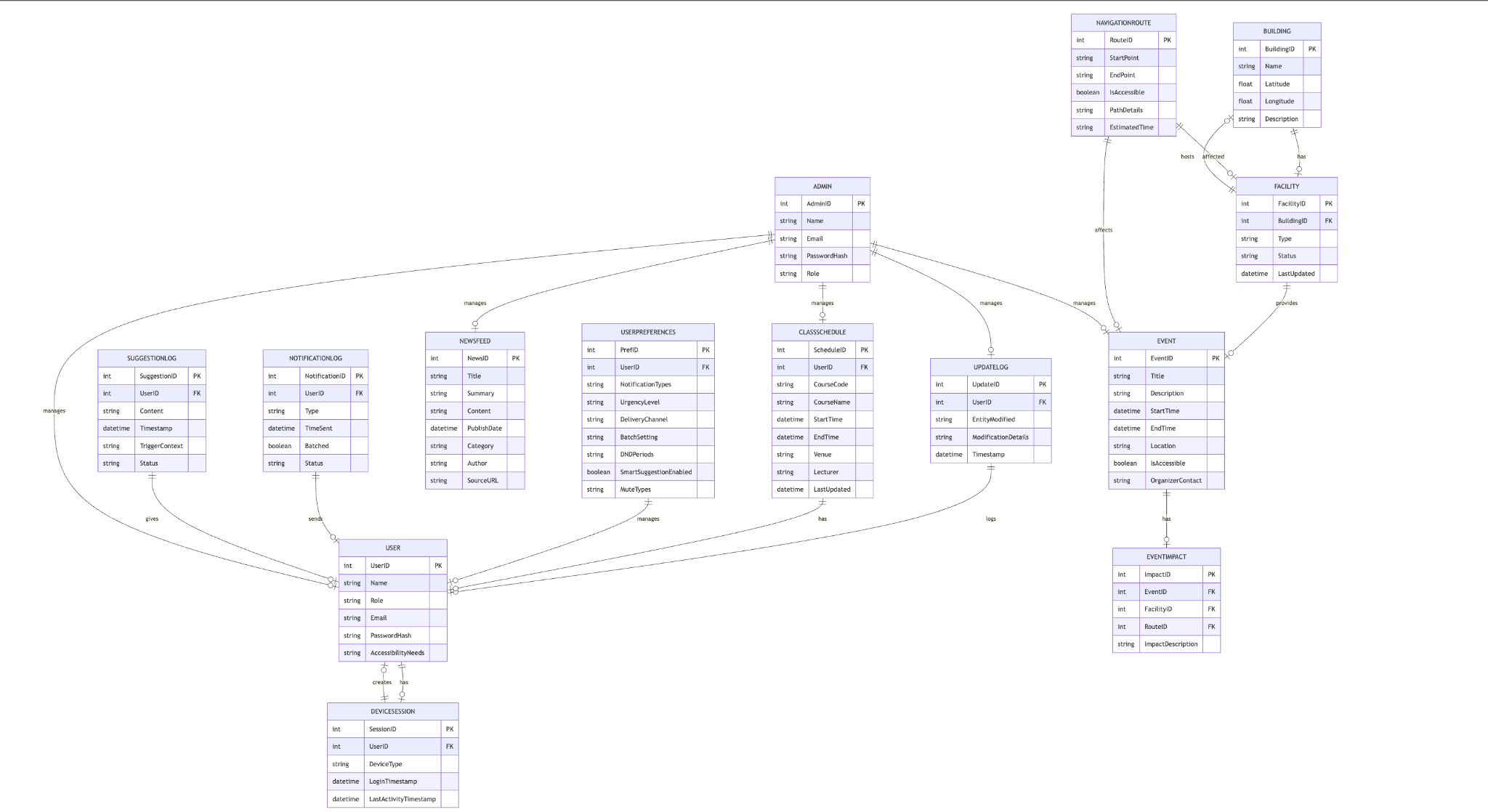
1. **Many-to-One**:

* Multiple Facilities belong to a single Building.
* Multiple EventImpact records may point to one Event.
* Multiple UpdateLog records may point to one Admin.

1. **Optional Relationships**:

* An EventImpact may affect either a Facility, a NavigationRoute, or both.
* AccessibilityNeeds may be empty for general users without specific impairments.

### [Entity Relationship Diagram](https://drive.google.com/file/d/128s2uW3bc4ZwtKK-0dMi5ypiuBF_y9JV/view?usp=sharing)



*Figure 3.5.1* [*Entity Relationship Diagram*](https://drive.google.com/file/d/128s2uW3bc4ZwtKK-0dMi5ypiuBF_y9JV/view?usp=sharing)

This section provides a high-level overview of the MMUAccess system's conceptual data model, as represented by the Entity-Relationship Diagram (ERD)**.** This model outlines the key entities within the system and their relationships, supporting the functional and non-functional requirements detailed in this document.

The Entity-Relationship Diagram (ERD) defines the logical architecture and data relationships that underpin the platform’s functionality. It encapsulates the essential entities, their attributes, and the interconnections required to support a smart, accessible, and responsive university navigation system.

## 3.6 Design Constraints

These constraints ensure the MMUAccess system is secure, functional, and maintainable. Key aspects include **strict security and privacy compliance** (PDPA, encrypted user preferences, compliant third-party notification services, user-controlled suggestions), **operational requirements** (active network connection for real-time features, administrative maintainability of data, use of an ACID-compliant RDBMS, and robust handling of third-party API reliability for class schedules and news feeds), and **flexible notification management** (modular and configurable batching logic).

### 3.6.1 Platform Compatibility

|  |  |
| --- | --- |
| **Field** | **Detail** |
| Requirement ID | REQ\_D001: Platform Compatibility |
| Description | The system shall be implemented as a responsive web and mobile application, functioning consistently across smartphones, tablets, and desktop computers using standard-compliant web technologies. |
| Constraint Type | Platform |
| Rationale | Ensures all users can access the system regardless of device, supporting inclusivity and reach. |
| Acceptance Criteria | Usability testing on a representative range of devices confirms correct functionality and appearance. |

### 

### 3.6.2 Accessibility Standards

|  |  |
| --- | --- |
| **Field** | **Detail** |
| Requirement ID | REQ\_D002: Accessibility Standards Compliance |
| Description | The user interface shall comply with WCAG 2.1 Level AA to ensure accessibility for users with visual, auditory, or motor impairments. |
| Constraint Type | Regulatory |
| Rationale | Aligns with international standards and legal requirements, supporting users with disabilities. |
| Acceptance Criteria | Accessibility audit reports show compliance with WCAG 2.1 Level AA. |

### 3.6.3 Integration Requirements

|  |  |
| --- | --- |
| **Field** | **Detail** |
| Requirement ID | REQ\_D003: Integration Requirements |
| Description | The system shall integrate with the MMU event calendar system and campus facilities management database, adhering to each system’s data formats, API specifications, and authentication protocols. |
| Constraint Type | Integration/Interoperability |
| Rationale | Enables real-time data synchronization and ensures compatibility with existing campus systems. |
| Acceptance Criteria | Successful integration testing; system can fetch, update, and display data from external APIs as specified. |

### 

### 3.6.4 Technology Constraints

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_D004: Technology Stack Constraints |
| **Description** | The frontend shall be built using HTML5, CSS3, and JavaScript (preferably React). The backend shall use Node.js or other approved server-side technology. Communication must use RESTful APIs. |
| **Constraint Type** | Technology/Implementation |
| **Rationale** | Aligns with university IT standards and available expertise; ensures maintainability and scalability. |
| **Acceptance Criteria** | Technology choices are documented; codebase reflects specified stack. |

### 3.6.5 Security and Privacy Compliance

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_D005: Security and Privacy Compliance |
| **Description** | The system shall comply with the Personal Data Protection Act (PDPA) of Malaysia, using secure password hashing and session management. |
| **Constraint Type** | Regulatory/Security |
| **Rationale** | Protects sensitive user data and meets legal obligations. |
| **Acceptance Criteria** | Security audit confirms PDPA compliance and no major vulnerabilities. |

### 

### 3.6.6 Network Dependency

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_D006: Network Dependency |
| **Description** | The application shall require an active internet connection for real-time features such as live updates and navigation recalculation. |
| **Constraint Type** | Operational |
| **Rationale** | Real-time data exchange is essential for system functionality. |
| **Acceptance Criteria** | System gracefully handles network loss, and displays user-friendly error messages. |

### 3.6.7 Maintainability

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_D007: Maintainability |
| **Description** | Administrative staff shall be able to maintain and update event and facility data through a secure admin interface, without developer intervention. |
| **Constraint Type** | Maintainability/Usability |
| **Rationale** | Reduces system downtime and maintenance costs. |
| **Acceptance Criteria** | Admin interface enables authorized users to manage content as specified. |

### 3.6.8 Database Constraints

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_D008: Database Constraints |
| **Description** | The system shall use an RDBMS (e.g., MySQL, PostgreSQL) supporting ACID properties and structured queries. |
| **Constraint Type** | Data Management |
| **Rationale** | Ensures data integrity, reliability, and consistency. |
| **Acceptance Criteria** | Database setup and testing confirm compliance with ACID principles. |

### 3.6.9 Third-Party API Reliability (for Class Schedule & News Feed)

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_D009: Third-Party API Reliability (for Class Schedule & News Feed) |
| **Description** | The system shall rely on stable, documented, and officially authorized APIs for class schedule and news feed integration, with clear error-handling for service downtime. |
| **Constraint Type** | Integration |
| **Rationale** | Ensures reliability of timetable/news functions even if external services are temporarily unavailable. |
| **Acceptance Criteria** | The app notifies users if class/news data cannot be fetched, and logs errors for troubleshooting. |

### 

### 3.6.10 User Notification Preferences Security

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_D010: User Notification Preferences Security |
| **Description** | All user notification filter preferences and settings must be securely stored, encrypted at rest, and protected against unauthorized access or tampering. |
| **Constraint Type** | Security/Privacy |
| **Rationale** | Prevents data leaks or manipulation of user-specific notification settings. |
| **Acceptance Criteria** | Security audit confirms that preferences are properly encrypted and only accessible to the owning user. |

### 3.6.11 Notification Delivery Platform Compliance

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_D011: Notification Delivery Platform Compliance |
| **Description** | Any third-party notification delivery service used must comply with PDPA and university privacy policies, ensuring no personal data is processed outside approved jurisdictions. |
| **Constraint Type** | Regulatory/Operational |
| **Rationale** | Maintains data privacy and legal compliance for notification-related integrations. |
| **Acceptance Criteria** | Service provider contracts reviewed and approved by university IT/security office. |

### 3.6.12 Real-Time Suggestion Accuracy

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_D012: Real-Time Suggestion Accuracy |
| **Description** | Smart suggestions and context-aware alerts must be based only on user-authorized data, and the logic must allow users to opt out of such features at any time. |
| **Constraint Type** | User Control/Privacy |
| **Rationale** | Respects user consent and avoids misuse of location, timetable, or behavior data. |
| **Acceptance Criteria** | Usability testing confirms opt-out/disable feature works as intended. |

### 3.6.13 Notification Batching Logic

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_D013: Notification Batching Logic |
| **Description** | The notification batching and fatigue prevention logic must be modular, easily updated, and support configuration by system administrators without redeployment. |
| **Constraint Type** | Maintainability/Implementation |
| **Rationale** | Allows fast tuning of batching rules and notification limits as user needs change. |
| **Acceptance Criteria** | Admins can update batching settings through config files or admin panel; no code change required. |

## 3.7 Software System Attributes

### 3.7.1 Availability

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA001: Availability |
| **Description** | MMUAccess shall be accessible 24/7, with planned maintenance windows announced in advance. The system shall maintain at least 99.5% uptime during university operating hours. |
| **Attribute Type** | Availability |
| **Rationale** | Ensures users can reliably access the system, reducing disruptions to campus navigation. |
| **Acceptance Criteria** | Monthly uptime reports confirm ≥99.5% availability (excluding scheduled maintenance). |

### 3.7.2 Security

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA002: Security |
| **Description** | All communication shall use HTTPS; only authorized university staff may update facility and event data. User data shall be encrypted and access controls enforced. |
| **Attribute Type** | Security |
| **Rationale** | Protects sensitive information, prevents unauthorized access, and ensures compliance with security standards. |
| **Acceptance Criteria** | Security audits reveal no critical vulnerabilities; data is encrypted at rest and in transit. |

### 3.7.3 Accessibility

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA003: Accessibility |
| **Description** | The system shall comply with WCAG 2.1 Level AA, supporting high contrast themes, screen readers, and keyboard navigation for users with disabilities. |
| **Attribute Type** | Accessibility |
| **Rationale** | Ensures inclusivity and legal compliance for users with varying abilities. |
| **Acceptance Criteria** | Accessibility audits confirm compliance; user testing by individuals with disabilities verifies feature usability. |

### 3.7.4 Reliability

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA004: Reliability |
| **Description** | The system shall achieve at least 99.5% uptime during university operating hours, and automatically resume services and data synchronization after connectivity restoration. Minor data failures or outages shall be handled gracefully, with user-friendly error messages and auto-recovery where possible. |
| **Attribute Type** | Reliability |
| **Rationale** | Reduces user frustration and loss of service due to system errors or network issues. |
| **Acceptance Criteria** | System logs confirm downtime and recovery periods; monthly uptime and recovery tests meet targets. |

### 3.7.5 Maintainability

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA005: Maintainability |
| **Description** | The system shall maintain at least 99.5% uptime and automatically recover from minor failures or outages within 10 minutes of connectivity restoration. The codebase shall be designed for ease of modification, enhancement, and bug fixing, allowing it to efficiently adapt to evolving user needs, new features, and system integrations. This includes adherence to established coding standards, a modular architecture, and comprehensive documentation. |
| **Attribute Type** | Maintainability |
| **Rationale** | Ensuring maintainability reduces user frustration and service disruptions caused by system or network issues. It also enhances the long-term viability and cost-efficiency of the system by minimizing the time and effort required for future development, updates, and maintenance activities. |
| **Acceptance Criteria** | 1. Code Quality: Code reviews and static analysis tools confirm adherence to defined coding standards (e.g., linting rules), with at least 90% of the codebase following these standards and supporting a modular architecture. 2. Documentation: Key system components and APIs must be fully documented according to internal standards and verified through documentation audits. 3. Efficiency: The average time to implement a minor bug fix (e.g., a UI text change) shall not exceed 2 person-hours. 4. Monitoring: System logs must confirm downtime and recovery periods. Monthly uptime and recovery tests should meet the specified targets. |

### 3.7.6 Portability

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA006: Portability |
| **Description** | The MMUAccess system shall operate seamlessly across modern web browsers (Chrome, Firefox, Safari, Edge) and mobile operating systems (iOS, Android), leveraging responsive design to support a wide range of screen sizes and devices.  The system’s database and backend components shall be designed for portability, enabling deployment and operation across different server environments (e.g., Linux, Windows Server) or distinct cloud platforms (e.g., AWS, Azure, GCP) with minimal configuration or code changes. |
| **Attribute Type** | Portability |
| **Rationale** | Portability ensures broad accessibility and compatibility for all campus users. It also allows for flexible deployment and hosting options, reduces vendor lock-in, and simplifies migration or disaster recovery across different infrastructure environments—contributing to the system’s long-term adaptability and resilience. |
| **Acceptance Criteria** | 1. The system, including its backend and database services, must be successfully deployed and fully operational on at least two distinct server environments or cloud platforms (e.g., a development server and a staging cloud environment) with less than 1 day or 8 person-hours of migration effort. This must be verified through deployment logs and post-deployment testing. 2. No core application code changes shall be required for migration between the specified environments, ensuring true cross-platform compatibility. |

### 

### 3.7.7 Scalability

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA007: Scalability |
| **Description** | The system shall support scaling to accommodate increased user load or new feature modules with minimal downtime. |
| **Attribute Type** | Scalability |
| **Rationale** | Ensures the platform remains performant as the university population or requirements grow. |
| **Acceptance Criteria** | System can be upgraded to support double the initial user capacity (e.g., from 500 to 1000 concurrent users) without major architectural changes or more than 1 hour of planned downtime. |

### 3.7.8 Auditability

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA008: Auditability |
| **Description** | The system shall log all critical administrative actions (e.g., event/facility updates, user role changes) for at least 1 year, and provide authorized users with access to audit logs. |
| **Attribute Type** | Auditability |
| **Rationale** | Supports troubleshooting, security investigations, and compliance. |
| **Acceptance Criteria** | Audit logs are available for inspection and meet data retention policies. |

### 

### 3.7.9 Performance

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA009: Performance |
| **Description** | The system shall respond to user actions within 3 seconds for 95% of requests, as specified in section 3.2. |
| **Attribute Type** | Performance |
| **Rationale** | Ensures good user experience and supports campus operational needs. |
| **Acceptance Criteria** | System logs and tests confirm average and percentile response times. |

### 3.7.10 Class Schedule Accuracy

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA010: Class Schedule Accuracy |
| **Description** | The class schedule feature shall maintain 99% accuracy and real-time synchronization with the university timetable system. |
| **Attribute Type** | Accuracy/Reliability |
| **Rationale** | Ensures students receive timely and correct schedule updates. |
| **Acceptance Criteria** | Audit logs confirm that class schedule data matches the source system in 99% of random samples. |

### 

### 3.7.11 News Feed Timeliness

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA011: News Feed Timeliness |
| **Description** | The campus news feed shall display announcements and news within 5 minutes of official release. |
| **Attribute Type** | Timeliness |
| **Rationale** | Ensures users have prompt access to the latest campus information. |
| **Acceptance Criteria** | Testing confirms news items appear in app within 5 minutes of backend update. |

### 3.7.12 Notification Filter Effectiveness

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA012: Notification Filter Effectiveness |
| **Description** | The system shall deliver only the notifications that match user-set filters, with a false positive/negative rate below 2% in usability tests. |
| **Attribute Type** | Accuracy/Effectiveness |
| **Rationale** | Maximizes relevance of notifications and prevents unwanted interruptions. |
| **Acceptance Criteria** | Controlled tests show ≤2% notifications delivered do not match user filter preferences. |

### 

### 3.7.13 Suggestion Relevance

|  |  |
| --- | --- |
| **Field** | **Detail** |
| **Requirement ID** | REQ\_SA013: Suggestion Relevance |
| **Description** | Smart suggestions and context-aware alerts must be relevant to user context at least 95% of the time, as measured in pilot user studies. |
| **Attribute Type** | Relevance |
| **Rationale** | Ensures suggestions are genuinely helpful and not ignored as spam. |
| **Acceptance Criteria** | Pilot user surveys rate relevance at 95% or above. |

## 

## 3.8 Supporting Information

### 3.8.1 References and Background information that supports the development of MMUAccess

* **MMU Campus Map and Facilities Management Data**
* This dataset is integral to the development of the *Campus Navigation* feature and the construction of the *facility database structure*, enabling efficient location-based services within the application.
* **MMU Official Event Calendar API Documentation**
* The API documentation is utilized for the seamless integration of *event-related functionalities*, directly supporting the system's event management and scheduling requirements.
* **WCAG 2.1 Accessibility Guidelines**
* Available at [W3C WCAG 2.1](https://www.w3.org/WAI/WCAG21/quickref/), these guidelines are employed to establish the *accessibility standards* and *user interface design principles*, ensuring that MMUAccess is fully accessible to a diverse range of users, including those with disabilities.
* **IEEE 830-1998 Standard for SRS Structure**
* This widely recognized standard provides the foundational framework for the *System Requirements Specification (SRS)*, ensuring that the document structure is well-organized and that the project requirements are formally defined and consistent.
* **Student Feedback from Questionnaires and Interviews**
* Insights collected from student questionnaires and interviews during the *elicitation stage* have been instrumental in refining the *functional* and *usability requirements*, directly influencing the design decisions to ensure the system effectively meets user needs.

### 

### 

### 3.8.2 Validation Sessions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Session ID** | **Date & Time** | **Technique** | **Section Reviewed** | **Participant & Role** | **No. of Defect** |
| VS01 | 2025-06-01 20:00 | Inspection | 3.1-3.3 Functions/Performance | Sim Boon Xun (Leader, Recorder), Hong Chia Qian (Reviewer, Negotiator) | 4 |
| VS02 | 2025-06-02 21:00 | Walkthrough | 3.4-3.6 Interface/Constraints | Sim Boon Xun (Leader, Recorder), Hong Chia Qian (Reviewer, Negotiator) | 3 |
| VS03 | 2025-06-09 20:00 | Walkthrough | 3.1-3.2 New Requirement added | Sim Boon Xun (Leader, Recorder), Hong Chia Qian (Reviewer, Negotiator) | 5 |
| VS04 | 2025-06-12 20:00 | Walkthrough | 3.2-3.7 add some point depends on New Requirement added | Sim Boon Xun (Leader, Recorder), Hong Chia Qian (Reviewer, Negotiator) | 6 |
| VS05 | 2025-06-18 13:00 | Walkthrough | Full document formatting and final content polishing; confirmed wording clarity, slide preparation, and presentation deadline | Sim Boon Xun (Leader, Recorder), Hong Chia Qian (Reviewer, Negotiator), Wang Kuang Wei (Recorder, Reviewer) | 9 |

### 3.8.3 Defect Summary

#### A. Content Defect

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req ID** | **Validation and Defect Description** | **Detected By** | **Comment/Suggested Fix** | **Session ID** | **Severity (1–5)** |
| REQ\_F007 | Missing user obstacle reporting feature. | Sim Boon Xun | Add new requirement and use case for feedback. | VS01 | 5 |
| REQ\_F007 | Missing class schedule integration for personalized routes. | Sim Boon Xun | Add new requirement for class schedule integration. | VS03 | 5 |
| REQ\_F008 | Campus news feed integration missing from main UI. | Sim Boon Xun | Add campus news feed integration requirement. | VS03 | 4 |
| REQ\_F009 | Users cannot customize notification preferences/filters. | Sim Boon Xun | Add customizable notification filter requirement. | VS03 | 3 |
| REQ\_F010 | Lack of context-aware smart suggestions/reminders. | Sim Boon Xun | Add smart suggestions/context-aware alerts requirement. | VS03 | 3 |
| REQ\_F011 | No notification fatigue prevention controls for users. | Sim Boon Xun | Add notification fatigue prevention (batching, mute, DND) requirement. | VS03 | 3 |
| REQ\_F008, REQ\_F009, REQ\_F011 | Unclear or ambiguous user interface specifications. | Hong Chia Qian | Redefined and clarified user interface specifications. | VS02 | 3 |
| N/A (Impacts underlying data model for all functional requirements) | Inaccurate or incomplete ERD definition and relationships. | Hong Chia Qian | Updated ERD definition, core entities, and key relationships for accuracy. | VS02, VS05 | 3 |
| N/A (Affects overall system design) | Missing or generic design constraints (software, hardware, regulatory, development). | Hong Chia Qian | Updated with detailed software, hardware, regulatory, and development constraints. | VS02 | 3 |
| N/A (Verification methodologies) | Missing security and performance testing sections in verification plan. | Wang Kuang Wei | Added new sections for Security Testing and Performance Testing. | VS04, VS05 | 3 |

#### B. Documentation Defect

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Page No.** | **Validation and Defect Description** | **Detected By** | **Comment/Suggested Fix** | **Session ID** | **Severity (1–5)** |
| 12 | Use case diagram not inserted; caption missing. | Sim Boon Xun | Add diagram and proper caption. | VS02 | 3 |
| 13-18 | Use case/state diagrams for new requirements missing | Sim Boon Xun | Add diagrams and proper captions | VS03 | 3 |
| 4 | Definition isn't arrange in alphabetical order | Hong Chia Qian | Rearrange the definition according to alphabetical order | VS05 | 4 |
| 6 | Lack of references | Hong Chia Qian | Added new external and internal document references. | VS05 | 3 |
| 7 | Use case diagram lack of info and diagram described wrongly | Hong Chia Qian | Created a new Use Case Diagram with updated use cases and added diagram description. | VS05,VS03 | 3 |
| 24 | Lack of definition | Wang Kuang Wei | Provided a new definition. | VS05 | 3 |
| N/A | Lack of definition | Hong Chia Qian | Added detailed definition. | VS05 | 3 |
| 25 | Unorganised alphabetical order and unrefined content | Wang Kuang Wei | Rearranged for alphabetical order and refined content (requires final check for scope). | VS05 | 4 |

#### C. Agreement Defect

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req ID** | **Validation Description/Stakeholder Concern** | **Mismatch** | **Detected By** | **Session ID** | **Severity (1–5)** |
| REQ\_F004 | Event filtering didn’t match student needs for category. | Filtering options | Sim Boon Xun | VS01 | 4 |
| REQ\_F009 | Users want finer control over notification types and urgency | Lacks customization | Sim Boon Xun | VS03 | 4 |
| REQ\_F011 | Users dissatisfied with notification overload; need mute/batch | No fatigue prevention | Sim Boon Xun | VS03 | 4 |

### 

### 3.8.4 Conflict Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Conflict ID** | **Conflict Description** | **Conflict Analysis** | **Stakeholders Involved** | **Session ID** |
| C01 | Should the system support obstacle reports? | Initial scope prioritized static navigation, creating a conflict with strong user need for dynamic environment awareness to ensure accurate and accessible navigation. | Team, Users, Simulated Stakeholders | VS01 |
| C02 | Should we include class schedule and news feed integration from our own SRS? | Initial project scope prioritized core navigation. However, strong student feedback emphasized the need for centralized information access, creating a conflict between the defined MVP and broader user experience expectations. | Team, users | VS03 |
| C03 | Should news feed be prioritized over events? | Debate arose over limited UI space on the homepage, leading to a prioritization conflict between presenting immediate news updates versus displaying upcoming events prominently. | Team, users | VS03 |
| C04 | How granular should notification filters be? | A conflict existed between developers' preference for simpler, easier-to-implement notification filters and users' strong desire for highly granular control (e.g., category, urgency, channel) to prevent notification fatigue. | Team, users, devs | VS03 |

### 

### 3.8.5 Conflict Resolution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Conflict ID** | **Conflict Resolution Strategy** | **Resolved (Y/N)** | **Outcome** | **Justification** |
| C01 | Facilitated team workshop with user representatives; prioritized feature based on critical user feedback regarding real-world campus navigation challenges. | Y | Added user obstacle reporting feature and its corresponding use case. | Decision driven by strong stakeholder demand (users) for real-time environmental awareness, recognizing its high impact on accessibility and the overall accuracy of the navigation system, and to mitigate potential user frustration. |
| C02 | Re-evaluation of initial scope through stakeholder negotiation sessions informed by comprehensive student feedback on the importance of centralized information access. | Y | Added Class Schedule and News Feed integration as core features. | Justified by overwhelming student demand for a centralized information hub. Analysis indicated this feature would significantly enhance daily campus navigation and information access, preventing fragmented user experiences and driving higher user adoption. |
| C03 | Development and presentation of multiple UI design mockups; conducted stakeholder feedback sessions and informal usability tests to gather preferences on homepage layout. | Y | Homepage design now accommodates both news and events feeds with clear visual distinction. | Resolution based on user feedback and usability tests confirming that a dual-feed approach provides the most comprehensive and user-friendly experience, allowing users to quickly access both types of critical campus information. |
| C04 | Iterative prototyping and user surveys during validation (VS03) to understand user preferences for notification control granularity. | Y | Implemented flexible notification filters in user settings, allowing customization by category, urgency, and channel. | Strong user demand, particularly to prevent notification fatigue, justified the added complexity of granular filters. This solution empowers users with personalized control, leading to increased user satisfaction and sustained engagement with notifications. |

### 3.8.6 Change Log

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Change ID** | **Req ID** | **Summary of Change** | **Proposed By** | **Date** | **Session ID** |
| CH01 | REQ\_F007 | Added obstacle reporting requirement | Sim Boon Xun | 2025-06-01 | VS01 |
| CH02 | REQ\_F004 | Enhanced event filtering options | Sim Boon Xun | 2025-06-01 | VS01 |
| CH03 | REQ\_F001–F006 | Refactored 3.1 Functions section with IDs, fields, and acceptance criteria | Sim Boon Xun | 2025-06-02 | VS01 |
| CH04 | REQ\_P001–P010 | Refactored 3.2 Performance Requirements with full attributes and test criteria | Sim Boon Xun | 2025-06-03 | VS01 |
| CH05 | N/A | Refined 3.8 Supporting Info, expanding references and context | Sim Boon Xun | 2025-06-04 | VS01 |
| CH06 | REQ\_SA006–SA013 | Enhanced 3.7 System Attributes with measurable criteria | Sim Boon Xun | 2025-06-04 | VS01 |
| CH07 | REQ\_D001–D013 | Expanded 3.6 Design Constraints with detailed fields | Sim Boon Xun | 2025-06-04 | VS01 |
| CH08 | N/A | Enhanced 3.5 Logical Database: added Admin role, ERD caption, entity integrity | Sim Boon Xun | 2025-06-04 | VS01 |
| CH09 | REQ\_I001–I020 | Refactored 3.4 Interface Requirements with ID structuring and measurables | Sim Boon Xun | 2025-06-04 | VS02 |
| CH10 | REQ\_U001–U012 | Refactored 3.3 Usability Requirements with attributes and criteria | Sim Boon Xun | 2025-06-04 | VS02 |
| CH11 | REQ\_F007 | Added class schedule integration | Sim Boon Xun | 2025-06-08 | VS03 |
| CH12 | N/A | Added use case/state diagrams for new requirements | Sim Boon Xun | 2025-06-09 | VS03 |
| CH13 | REQ\_F008 | Added campus news feed integration | Sim Boon Xun | 2025-06-13 | VS03 |
| CH14 | REQ\_F009 | Added customizable notification filters | Sim Boon Xun | 2025-06-13 | VS03 |
| CH15 | REQ\_F010 | Added smart suggestions/context-aware alerts | Sim Boon Xun | 2025-06-13 | VS03 |
| CH16 | REQ\_F011 | Added notification fatigue prevention | Sim Boon Xun | 2025-06-13 | VS03 |
| CH17 | N/A | Applied consistent heading and section numbering (H1–H3) across document | Sim Boon Xun | 2025-06-17 | VS05 |
| CH18 | N/A | Added section 4.3 Stakeholder and Verification Summary | Sim Boon Xun | 2025-06-17 | VS05 |
| CH19 | N/A | Added initial Requirements Traceability Matrix to 3.9 | Sim Boon Xun | 2025-06-17 | VS05 |
| CH20 | N/A | Formatted and finalized this Change Log | Sim Boon Xun | 2025-06-17 | VS05 |
| CH21 | N/A | Clarified Purpose and Scope in Sections 1.1 and 1.2 | Hong Chia Qian | 2025-06-10 | VS04 |
| CH22 | N/A | Updated SRS Sections 3.4 to 3.7 with new requirements; migrated to Google Docs | Sim Boon Xun | 2025-06-12 | VS04 |
| CH23 | N/A | Edited 3.5 Logical DB to include Admin entity and relationships | Hong Chia Qian | 2025-06-16 | VS05 |
| CH24 | N/A | Updated ERD (section 3.5) with Admin role | Hong Chia Qian | 2025-06-16 | VS05 |
| CH25 | N/A | Refined sections 4.3, 5.1, 5.2 (Assumptions, Acronyms, etc.) | Wang Kuang Wei | 2025-06-17 | VS05 |
| CH26 | N/A | Refined 3.7, 4.1, 4.2 (System Attributes, Traceability, Testing) | Hong Chia Qian | 2025-06-17 | VS05 |
| CH27 | N/A | Updated 3.4–3.6 Interface, ERD, and Design Constraint sections | Hong Chia Qian | 2025-06-17 | VS05 |
| CH28 | N/A | Alphabetized Definitions, added new references, created use case diagram | Wang Kuang Wei | 2025-06-17 | VS05 |

### 

### 3.8.7 Requirements Traceability Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirement ID** | **Requirement Description** | **Linked Goal(s)** | **Feature(s)** | **Use Case(s)** | **Traceability Score** |
| REQ\_F001 | User Registration/Login | Account Access | User Auth | UC001 | 100 |
| REQ\_F002 | View Campus Events | Event Awareness | Event List | UC002 | 100 |
| REQ\_F003 | Display Event Details | Event Awareness | Event Details | UC002 | 100 |
| REQ\_F004 | Event Filtering | Event Personalization | Filter Events | UC002 | 90 |
| REQ\_F005 | Accessible Route Guidance | Accessibility | Route Finder | UC003 | 100 |
| REQ\_F006 | No Events Message | User Experience | Event List | UC002 | 100 |
| REQ\_F007 | Class Schedule Integration | Schedule Convenience | Class Schedule | UC007 | 100 |
| REQ\_F008 | Campus News Feed Integration | Information Access | News Feed | UC008 | 100 |
| REQ\_F009 | Customizable Notification Filters | User Personalization | Notification Filter | UC009 | 95 |
| REQ\_F010 | Smart Suggestions/Context-Aware Alerts | Proactive Assistance | Smart Suggestions | UC010 | 95 |
| REQ\_F011 | Notification Fatigue Prevention | User Experience | Notification Fatigue Prevention | UC011 | 95 |

### 3.8.8 Role in Requirements Validation, Negotiation & Management

|  |  |  |
| --- | --- | --- |
| **Student Name** | **Primary Responsibility** | **No. of Sessions Participated** |
| Sim Boon Xun | Leader, Facilitator | 5 |
| Hong Chia Qian | Reviewer, Negotiator | 5 |
| Wang Kuang Wei | Recorder, Reviewer | 1 |
| Chia Kok Ang | Negotiator, Reviewer | 0 |

### 3.8.9 Version Control & Configuration Summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **Commits** | **Pull Requests** | **Change Log Entries** |
| Sim Boon Xun | 11 | 0 | 11 |
| Hong Chia Qian | 7 | 10 | 7 |
| Wang Kuang Wei | 2 | 0 | 2 |
| Chia Kok Ang | 0 | 0 | 0 |

### 3.8.10 Supporting Document

#### 

This section includes links to key artefacts that support the development of this Software Requirements Specification (SRS). The supporting materials are stored in a shared Google Drive folder and consist of:

#### 

The Entity-Relationship Diagram (ERD) that visualizes the system’s data structure.

The version history of the original SRS draft developed using Google Docs, which provides transparency into the evolution of the document and collaborative contributions.

#### 

Supporting Documents Folder:  
<https://drive.google.com/drive/folders/1TZjdFZLX8PHqZuo5Ufv1BeArC_65TGtZ?usp=sharing>

# 4.0 Verification

This section details the comprehensive approach to verifying that the MMUAccess system meets all specified functional and non-functional requirements. Verification will be conducted through a combination of systematic reviews, rigorous testing, and stakeholder validation.

## 4.1 Requirement Traceability

To ensure comprehensive coverage and validation, each requirement will be traced to its corresponding test cases) within a Requirements Traceability Matrix. This matrix will confirm that every requirement is tested and validated throughout the development lifecycle.

## 4.2 Verification Techniques

This table outlines the primary methods used for verifying the MMUAccess system requirements:

|  |  |
| --- | --- |
| **Verification Method** | **Description** |
| Requirement Reviews | Thorough stakeholder and peer reviews to confirm clarity and completeness. |
| Functional Testing | Test cases to verify how the system reacts to user input and actions. |
| Usability Testing | Target consumers evaluate user interfaces with an emphasis on accessibility. |
| Acceptance Testing | End-user verification that the system satisfies all performance and functional requirements. |
| Integration Testing | Interactions between the Facilities DB, Event API, and MMUAccess are verified. |
| Security Testing | Tests conducted to ensure the system protects data and maintains functionality as intended, safeguarding against vulnerabilities and unauthorized access. |
| Performance Testing | Tests to evaluate system responsiveness, stability, scalability, and resource usage under various workloads |

## 4.3 Stakeholder and Verification Summary

This section provides an overview of how each primary stakeholder group will contribute to and benefit from the verification process, along with a summary of the types of testing they are involved in.

| **Stakeholder** | **Verification Method** | **Verification Summary** |
| --- | --- | --- |
| Students | Functional Testing, Usability Testing, Acceptance Testing | Verify registration, event browsing, route generation, class schedule, notification filters, smart suggestions, accessibility compliance. |
| Staff (Admin) | Functional Testing, Integration Testing, Requirement Reviews | Verify event management, facility status update, database correctness, logging mechanism, and real-time updates for users. |
| Project Developer | Integration Testing, Security Testing, Performance Testing | Verify API integration, data synchronization, security compliance, scalability, and full requirement coverage based on traceability matrix. |

### 4.3.1 Students - Verification Test Cases

​​Verification Objective: Confirm the system meets usability, accessibility, and functional needs for student users.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Method** | **Expected Result** | **Status** |
| ST-UAT-01 | Verify user registration and login | Functional Testing | User registers and logs in successfully with valid credentials | Pass/Fail |
| ST-UAT-02 | Verify event list viewing | Functional Testing | System displays correct event list with all key info | Pass/Fail |
| ST-UAT-03 | Verify event detail viewing | Functional Testing | Full event info with accessibility tags shown correctly | Pass/Fail |
| ST-UAT-04 | Verify event filtering by date/type/accessibility | Functional Testing | Filtered event list returns correct results | Pass/Fail |
| ST-UAT-05 | Verify accessible route calculation | Functional Testing | Route considers accessibility needs, avoids obstacles | Pass/Fail |
| ST-UAT-06 | Verify class schedule integration | Functional Testing | Student class schedule shows correct data synced with timetable | Pass/Fail |
| ST-UAT-07 | Verify smart suggestion delivery | Functional Testing | Receive timely personalized alerts/reminders | Pass/Fail |
| ST-UAT-08 | Verify notification filter customization | Functional Testing | Only desired notifications are received as per settings | Pass/Fail |
| ST-UAT-09 | Verify notification fatigue prevention | Functional Testing | System batches/mutes non-urgent notifications | Pass/Fail |
| ST-UAT-10 | Verify accessibility compliance | Usability Testing | Users with disabilities can fully operate system features | Pass/Fail |

### 4.3.2 Admin (Staff) VerificationTest Cases

Verification Objective: Ensure administrative functions for event and facility management, logging, and data synchronization operate correctly and reliably.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Method** | **Expected Result** | **Status** |
| AD-UAT-01 | Verify event creation and editing | Functional Testing | Admin can add, edit, delete events correctly | Pass/Fail |
| AD-UAT-02 | Verify facility status update | Functional Testing | Admin can update facility availability; users get notified | Pass/Fail |
| AD-UAT-03 | Verify update logging | Functional Testing | All changes logged with timestamp and admin ID | Pass/Fail |
| AD-UAT-04 | Verify integration with event calendar | Integration Testing | Event data correctly syncs from calendar system | Pass/Fail |
| AD-UAT-05 | Verify facility data synchronization | Integration Testing | Facilities DB syncs real-time with correct update propagation | Pass/Fail |

### 

### 4.3.3 Project Developer Verification Test Case

Verification Objective: Validate that the system's API integrations, database structure, scalability, security, and traceability meet the defined project requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Method** | **Expected Result** | **Status** |
| PD-UAT-01 | Verify API integration for event & facility | Integration Testing | API fetches, updates, and syncs data correctly | Pass/Fail |
| PD-UAT-02 | Verify database structure & constraints | Requirement Review | ERD entities function correctly with integrity | Pass/Fail |
| PD-UAT-03 | Verify system scalability & performance | System Testing, Performance testing | System sustains up to 500 concurrent users smoothly | Pass/Fail |
| PD-UAT-04 | Verify security compliance (PDPA, HTTPS) | System Testing, Security Testing | User data fully encrypted, secure session management | Pass/Fail |
| PD-UAT-05 | Verify traceability coverage | Requirement Review | All requirements traceable to test cases and implementation | Pass/Fail |

# 5.0 Appendices

## 5.1 Assumptions and Dependencies

### 5.1.1 User Connectivity Prerequisite:

* ﻿﻿Assumption: It is assumed that all end-users of the MMUAccess system (including students, staff, and visitors) will possess and utilize devices equipped with a functional and stable internet connection. This is essential for accessing real-time system functionalities, receiving live updates, and ensuring comprehensive service availability.
* ﻿﻿Dependency: The consistent availability and reliability of internet infrastructure, both within the MMU campus environment and in external user locations, is a fundamental external dependency for system operability.

### 5.1.2 MMU Backend System Integration and Data Provision:

* ﻿﻿Assumption: The project assumes that MMU's existing core systems, specifically the official event calendar and campus facility management systems, will be fully operational, consistently accessible, and capable of providing real-time data access via robust Application Programming Interfaces (APIs). This includes comprehensive and accurate data pertaining to event schedules, venue details, and detailed facility attributes (e.g., accessibility tags, structural information).
* ﻿﻿Dependency: This initiative is critically dependent on the ongoing stability, uptime guarantees, and the provision of well-documented, secure, and performant APIs from MMU's Information Technology (IT) department and the respective system owners. Formal agreement on data exchange protocols and integration timelines is required.

### 5.1.2 Administrative Data Maintenance and Accuracy:

* ﻿﻿Assumption: It is an explicit assumption that designated MMU administrative personnel will consistently, accurately, and in a timely manner, update all relevant information pertaining to campus facilities and event-related infrastructure within the underlying MMU backend systems.  
  This encompasses updates on facility statuses (e.g., elevator malfunctions, ramp closures, temporary path obstructions) and details regarding ongoing construction zones.
* ﻿﻿Dependency: The accuracy and utility of MMUAccess's real-time features are directly dependent on the disciplined adherence to established data update protocols and the proactive engagement of MMU's Facilities Management, Event Coordination, and other relevant administrative teams.
* ﻿﻿Assumption: The system relies on the assumption that the foundational university map data, which will either be directly integrated or serve as a basis for MMUAccess's navigation features, is perpetually up-to-date, geographically accurate, and sufficiently detailed. This detail includes precise mapping of accessible pathways, building layouts, and key points of interest.
* ﻿﻿Dependency: This represents a critical dependency on the MMU departments responsible for campus spatial data management and mapping, requiring their continuous commitment to maintaining the currency and precision of all provided geographical information.

### 5.1.3 End-User Device and Browser Compatibility:

* ﻿﻿Assumption: It is assumed that end-users will access MMUAccess using computing devices (personal computers, laptops, smartphones, tablets) that are compatible with current web standards and contemporary browser versions. Furthermore, these devices are assumed to support common assistive technologies and accessibility tools (e.g., screen readers) as needed by users with diverse accessibility requirements.
* ﻿﻿Dependency: While the system will aim for broad compatibility, its optimal functionality and accessibility features are dependent on users maintaining updated operating systems, web browsers, and accessibility software on their personal devices.

## 5.2 Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| API | Application Programming Interface |
| DB | Database |
| HTTPS | Hypertext Transfer Protocol Secure |
| PDPA | Personal Data Protection Act (Example) |
| REQ | Requirement |
| SRS | Software Requirements Specification |
| UC | Use Case |
| UI | User Interface |
| UAT | User Acceptance Testing (Example) |
| WCAG | Web Content Accessibility Guidelines |