

**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 4](#_Toc198838884)

[1.1 Purpose 4](#_Toc198838885)

[1.2 Scope 4](#_Toc198838887)

[1.3 Product Overview 4](#_Toc198838888)

[1.3.1 Product Perspective 4](#_Toc198838889)

[1.3.2 Product Function 4](#_Toc198838890)

[1.3.3 User Characteristics 4](#_Toc198838891)

[1.3.4 Limitation 4](#_Toc198838892)

[1.4 Definition 4](#_Toc198838893)

[2.0 References 4](#_Toc198838894)

[3.0 Requirement 5](#_Toc198838895)

[3.1 Functions 5](#_Toc198838896)

[3.2 Performance Requirements 5](#_Toc198838897)

[3.3 Usability Requirement 5](#_Toc198838898)

[3.4 Interface Requirement 5](#_Toc198838899)

[3.5 Logical Database Requirement 5](#_Toc198838900)

[3.6 Design Constrains 5](#_Toc198838901)

[3.7 Software System Attributes 5](#_Toc198838902)

[3.8 Supporting Information 5](#_Toc198838903)

[4.0 Verification 5](#_Toc198838904)

[5.0 Appendices 5](#_Toc198838905)

[5.1 Assumptions and Dependencies 5](#_Toc198838906)

[5.2 Acronyms and Abbreviations 5](#_Toc198838907)

## 

# Introduction

## 1.1 Purpose

The purpose of this document is to specify the software requirements for MMUAccess, a campus accessibility navigation system. The document outlines the functionality, constraints, and design requirements of the platform, which aims to assist students, staff, and visitors in navigating the university campus — especially those with accessibility needs. This document is intended for software developers, system designers, project stakeholders, and testers involved in the development and deployment of the system.

## 1.2 Scope

MMUAccess shall facilitate mainly the following operations

1. Navigation across campus with accessible route
2. Real-time updates on campus events
3. Integration with MMU’s event calendar to guide users to accessible event locations

## 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:

i. Generate accessible navigation routes across the MMU campus, avoiding stairs, blocked paths, or other obstacles.  
ii. Provide real-time notifications and updates regarding campus events and their impact on navigation routes.  
iii. Integrate with MMU’s official event calendar to help users locate event venues and determine if the routes and locations are accessible.

### 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 |
| MMUAccess | The name of the proposed campus accessibility navigation system for MMU. |
| Accessible Route | A navigation route that avoids obstructions like stairs or blocked pathways in order to accommodate people with disabilities. |
| 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. |
| User | Any user of MMUAccess, including visitors, employees, and students. |
| Admin | The system's event and facility information is updated by authorised university staff. |
| WCAG | International guidelines for web accessibility are called Web Content Accessibility Guidelines. |
| 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. |
| Accessibility Tags | Metadata that indicates if a venue or event satisfies accessibility requirements. |

# 2.0 References

i. IEEE Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*.

# 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.

A diagram of a software system

AI-generated content may be incorrect.

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_F001 | **Version** | 1.0 |
| **Description** | System shall allow users to register for an account or log in using valid credentials. | | |
| **Author** | Naqib | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_F002 | **Version** | 1.0 |
| **Description** | System shall allow user to access and view a list of current events | | |
| **Author** | Naqib | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_F003 | **Version** | 1.0 |
| **Description** | System shall display event name, location, time, and accessibility info | | |
| **Author** | Naqib | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_F004 | **Version** | 1.0 |
| **Description** | System shall allow users to filter events based on date and category | | |
| **Author** | Naqib | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_F005 | **Version** | 1.0 |
| **Description** | System shall allow users to get accessible route guidance to event locations. | | |
| **Author** | Naqib | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_F006 | **Version** | 1.0 |
| **Description** | System shall display a message if no upcoming events are available | | |
| **Author** | Naqib | | |

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

## 3.2 Performance Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** | **Author** |
| REQ\_P001 | The system shall display event and route results within 3 **seconds** of request. | High | Naqib |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** | **Author** |
| REQ\_P002 | The system shall support **up to 500 concurrent users** (web and mobile) without performance lag | High | Naqib |

## 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.

* The system shall provide a user interface with intuitive navigation and minimal learning curve.
* Users shall be able to complete core tasks (e.g., searching for accessible routes, viewing event locations) with no more than 3 interactions (clicks/taps).
* The system shall offer multi-language support, with English and Bahasa Malaysia as defaults.
* Tooltips, onboarding guidance, and help sections shall be available to assist new users.
* A high-contrast visual mode and adjustable font sizes shall be included for visually impaired users.
* System usability shall comply with ISO 9241 and WCAG 2.1 standards, aiming for at least Level AA conformance.

## 3.4 Interface Requirement

3.4.1 User Interface

The system shall provide a responsive web interface compatible with Chrome, Firefox, Safari, and mobile browsers.

* Mobile interface shall be optimized for both iOS and Android devices.
* The UI shall follow consistent design patterns, with accessible components (buttons, input fields, modals) adhering to WCAG 2.1.
* The home page shall display navigation shortcuts, recent events, and campus alerts.
* Users shall be able to access features via touch, keyboard, or voice commands.

3.4.2 Hardware Interface

The system shall be operable on common smartphones, tablets, and desktop computers with standard web browsers.

* No special hardware is required beyond typical accessibility tools (e.g., screen readers, voice-to-text).

3.4.3 Software Interface

The system shall integrate with:

* MMU Event Calendar API – for retrieving event details and locations.
* MMU Facilities Management Database – for updated campus infrastructure data.
* All external integrations shall use secure RESTful APIs with JSON format.

3.4.4 Communications Interface

* All communication between client and server shall use HTTPS for secure transmission.
* The system shall be hosted on a cloud-based platform supporting real-time updates and reliable uptime.

## 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 students, staff, and visitors using the system. | UserID (PK), Name, Role (Student, Staff, Visitor), Email, PasswordHash, AccessibilityNeeds (e.g., wheelchair, visual aid) |
| **DeviceSession** | Stores session data for cross-platform login and usage. | SessionID (PK), UserID (FK), DeviceType, LoginTimestamp, LastActivityTimestamp |
| **Building** | Represents each building on campus. | BuildingID (PK), Name, Latitude, Longitude, Description |
| **Facility** | Details accessibility features of buildings and paths. | FacilityID (PK), BuildingID (FK), Type (Ramp, Elevator, Accessible Entrance), Status, LastUpdated |
| **NavigationRoute** | Stores pre-calculated routes across campus. | RouteID (PK), StartPoint, EndPoint, IsAccessible, PathDetails, EstimatedTime |
| **Event** | Represents campus events synced with MMU’s event calendar. | EventID (PK), Title, Description, StartTime, EndTime, Location, IsAccessible, OrganizerContact |
| **EventImpact** | Links events to impacted facilities or routes. | ImpactID (PK), EventID (FK), FacilityID (FK), RouteID (FK, optional), ImpactDescription |
| **UpdateLog** | Logs updates made by admin staff. | UpdateID (PK), UserID (FK), EntityModified, ModificationDetails, Timestamp |

*Table 3.5.1 Data Entities and Attributes*

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.

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

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

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

The Entity-Relationship Diagram (ERD) for MMUAccess illustrates the logical structure of the system’s database. It identifies the main entities involved—such as users, buildings, facilities, events, and navigation routes—and the relationships between them. This diagram serves as a blueprint for how data will be stored, connected, and retrieved within the system, supporting core functions like accessible navigation, real-time updates, and event impact tracking. The ERD ensures a consistent and scalable design that meets the accessibility and usability goals of the MMUAccess platform.

A diagram of a software application

Description automatically generated with medium confidence

*Figure 3.5.1 Entity Relationship Diagram*

Data Integrity and Constraints

- **Unique Constraints**

* Email in User must be unique.
* Primary keys must be unique and auto-incremented if needed.

- **Referential Integrity**

* Foreign keys ensure links between events, buildings, users, etc., are valid.

- **Value Constraints**

* Controlled vocabularies for Role, Type, Status using ENUMs.
* AccessibilityNeeds as a multi-select SET for flexibility.

- **Nullable Fields**

* FacilityID and RouteID in EventImpact may be NULL to indicate partial impacts.

- **Data Update Logging**

* All modifications to accessibility-related entities must be logged in UpdateLog.

Justification for Design

* The schema supports **real-time accessibility data** and **event routing** integration.
* EventImpact acts as a bridge between static data (routes, facilities) and dynamic updates (events).
* All data is normalized to reduce redundancy and maintain referential integrity.
* Role-based user modeling prepares for future expansion (e.g., admin dashboards, staff permissions).
* Ensures **mobile-first access** with lightweight queries (pre-computed routes, indexed foreign keys).

## 3.6 Design Constrains

Design constraints specify limitations on the design or implementation of the system due to:

* Software, hardware, or platform requirements,
* Regulatory standards,
* Development practices,
* System environment.

**Design Constrains**

**3.6.1 Platform Compatibility**

The system shall be implemented as a responsive web and mobile application. It must function consistently across multiple platforms and devices, including smartphones, tablets, and desktop computers, using standard-compliant web technologies.

**3.6.2 Accessibility Standards**

The user interface design must comply with the **Web Content Accessibility Guidelines (WCAG) 2.1**, ensuring that the system is usable by individuals with visual, auditory, or motor impairments.

**3.6.3 Integration Requirements**

MMUAccess shall integrate with the following existing university systems:

* The **official MMU event calendar system**, to fetch event details and update the accessibility status of events.
* The **campus facilities management database**, to retrieve real-time information about the availability and status of paths, elevators, and facilities.

All integrations must adhere to the respective systems’ data formats, API specifications, and authentication protocols.

**3.6.4 Technology Constraints**

* The frontend of the system shall be developed using **HTML5**, **CSS3**, and **JavaScript** (preferably using the **React** framework).
* The backend must be built using **Node.js** or any other server-side technology approved by MMU’s IT department.
* Communication between client and server components shall use **RESTful APIs**.

**3.6.5 Security and Privacy Compliance**

All personal and sensitive user data must be managed in compliance with the **Personal Data Protection Act (PDPA) of Malaysia**. The system shall employ secure password hashing algorithms and implement secure session management for user authentication.

**3.6.6 Network Dependency**

The application depends on an active internet connection to deliver real-time functionalities, including live navigation updates, event impact notifications, and route recalculations.

**3.6.7 Maintainability**

The system must be designed to allow authorized administrative staff to maintain and update event data and campus facility statuses through a secure admin interface without requiring developer intervention.

**3.6.8 Database Constraints**

A **relational database management system (RDBMS)** such as **MySQL** or **PostgreSQL** shall be used to ensure data integrity, support structured queries, and maintain ACID (Atomicity, Consistency, Isolation, Durability) compliance.

## 3.7 Software System Attributes

### 3.7.1 Availability MMUAccess will be accessible around-the-clock, particularly during school hours. The system will notify users in advance of scheduled maintenance windows.

### 3.7.2 Security HTTPS will be used by the system to ensure secure communication. Facility and event data can only be updated by authorised university employees. To stop unwanted changes, user data will be encrypted and access limits will be implemented.

### 3.7.3 Accessibility The system must be usable by people with visual, auditory, and motor disabilities in accordance with WCAG 2.1 (Web Content Accessibility Guidelines). High contrast themes, screen reader compatibility, and keyboard navigation are among the features.

### 3.7.4 Reliability With a goal uptime of 99.5%, the system must run continuously. When connectivity is restored, it should continue services and gracefully handle small data failures or outages.

### 3.7.5 Maintainability In order to facilitate future updates, such as new accessibility features or third-party integrations, the codebase will adhere to modular architecture and documentation standards.

### 3.7.6 Portability Through web browsers and mobile apps, MMUAccess will work on a variety of systems, including Windows, macOS, Android, and iOS. To accommodate various screen sizes and devices, the application will be created with responsive design.

## 3.8 Supporting Information

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

* MMU Campus Map and Facilities Management Data
* MMU Official Event Calendar API documentation
* WCAG 2.1 Accessibility Guidelines (<https://www.w3.org/WAI/WCAG21/quickref/>)
* IEEE 830-1998 Standard for SRS Structure
* Comments on accessibility from student questionnaires and interviews that were done during the elicitation stage

# 4.0 Verification

**4.0 Verification**

Reviews, inspections, testing, and user validation will all be used to confirm that the MMUAccess system is operating as intended and that all requirements have been met.

**4.1 Requirement Traceability**

To make sure every need has been met and validated during development, it will be connected to the test case that goes with it.

**4.2 Verification Techniques**

|  |  |
| --- | --- |
| **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. |

# 5.0 Appendices

## 5.1 Assumptions and Dependencies

* In order to use MMUAccess, users must have a functional internet connection.
* MMU's event calendar and facility systems will be up and running, including real-time API access.
* Admins will routinely update facility statuses and construction zones.
* The system makes the assumption that the university's map data is up-to-date and correct
* The online and mobile devices will be compatible with accessibility tools and current browsers.

## 5.2 Acronyms and Abbreviations

|  |  |
| --- | --- |
| Term | Defenition |
| MMU | Multimedia University |
| WCAG | Web Content Accessibility Guidelines |
| API | Application Programming Interface |
| SRS | Software Requirements Specification |
| DB | Database |
| UI | User Interface |
| HTTPS | Hypertext Transfer Protocol Secure |
| REQ | Requirement |
| UC | Use Case |