

**TRIMESTER March/April, 2025**

**CSE6224 Software Requirements Engineering**

**PROJECT 1**

**Campus Accessibility Navigation System with Facilities and Event Integration**

|  |  |  |
| --- | --- | --- |
| Name | Student ID | Course |
| Chia Kok Ang | 242UC24423 | Bachelor of Computer Science |
| Wang Kuang Wei | 242UC2451W | Bachelor of Computer Science |
| Sim Boon Xun | 242UC2451H | Bachelor of Computer Science |
| Hong Chia Qian | 1211107977 | Bachelor of Computer Science |

**Table of Content**

[**1. Introduction 3**](#_h8fquqki3ff0)

[1.1. Purpose 3](#_3jnzy4l4twfh)

[1.2. Scope 3](#_3se84addo5mn)

[1.3. Product overview 3](#_lnol6q7w8st4)

[1.3.1. Product perspective 3](#_hvm2mq5vq2sh)

[1.3.2. Product functions 3](#_yk0hmelp6gd0)

[1.3.3. User characteristics 4](#_baec0heznu0u)

[1.3.4. Limitations 4](#_1htcnmt61b5c)

[1.4. Definitions 5](#_gzcud1d4ink0)

[**2. References 6**](#_u4dqhd1xh2fh)

[**3. Requirement 7**](#_9spmyr10crjp)

[3.1. Functions 7](#_2yd6zlywmx06)

[3.1.1. Requirements Prioritization and Classification 7](#_5yw6fhijz4jb)

[● Basic Needs 7](#_mtzjowo3tssv)

[● Performance Needs 7](#_1dzb5w83b3f3)

[● Excitement Needs 8](#_plypieuijayk)

[● Indifferent Needs 8](#_iwjlyxu9tgxw)

[● Reverse Needs 9](#_lxunwh4eyagi)

[3.2. Performance requirements 9](#_b9y6p3ofs1r7)

[3.3. Usability requirements 9](#_qdys2ismd2r2)

[3.4. Interface requirements 9](#_bg01v5nqgphh)

[3.5. Logical database requirements 10](#_n1t0l9z3238a)

[3.5.1. Entity Relationship Diagram 11](#_j3eqjwglge6t)

[Diagram 3.1 Entity Relationship Diagram 11](#_imkqo6rv09tp)

[3.6. Design constraints 12](#_xl8sy6s7h9oh)

[3.7. Software system attributes 12](#_mhikl5e82jkj)

[3.8. Supporting information 12](#_w22ctkm8m5ga)

[3.9. Behavioral Models 13](#_x3nxkuboj331)

[3.9.1. Use Case Diagram 13](#_6dx6qkx6j18z)

[Diagram 3.2 Use Case Diagram 13](#_o4fwwb9zf4hz)

[3.9.2. State Chart Diagram 14](#_9ugqh91uurzk)

[Diagram 3.3 Student Interaction State Diagram 14](#_7v3bn9iyzdmt)

[Diagram 3.4 MMU Staff /Admin Interaction State Diagram 14](#_lubg2720zzax)

[Diagram 3.5 System Notification State Diagram 14](#_bskzt0sy8i2j)

[Diagram 3.6 System Notification State Diagram 15](#_tl7g9oijzta)

[Diagram 3.7 Data Synchronization State Diagram 15](#_h9vkjhjwuo9l)

[3.9.3. Activity Diagram 16](#_3i00iy4x4qjo)

[Diagram 3.8 User journey activity diagram 16](#_y8n7tdqh1onh)

[Diagram 3.9 admin workflow activity diagram 17](#_ml3k9tiuzapr)

[Diagram 3.10 Real time update and notification activity diagram 18](#_eplglwdpdwuw)

[Diagram 3.11 User exiting the navigation system activity diagram 19](#_e1r5cc80ae0q)

[Diagram 3.12 system failure activity diagram 19](#_mg0zjncjafep)

[**4. Verification 20**](#_hh6eos5qfdj2)

[4.1. Stakeholders and Verification Summary 20](#_au2h67eaoxvu)

[4.1.1 Students Verification 20](#_cua4r3fh8k1o)

[4.1.2 MMU Staff / Admin Verification 22](#_omyo7b8cy11v)

[4.1.3 Software Developers (Project Team) Verification 23](#_waqsmkbtltlq)

[4.2 Stakeholders and Verification Summary 24](#_q85z7ueb6a4s)

[Functional Testing 24](#_cua4r3fh8k1o)

[Performance Testing 24](#_cua4r3fh8k1o)

[Usability Testing 24](#_cua4r3fh8k1o)

[Security Verification 24](#_cua4r3fh8k1o)

[Interface Testing 24](#_cua4r3fh8k1o)

[Error Handling Testing 24](#_c8686p87kpzi)

[**5. Appendices 25**](#_n7g1fg9no8p7)

[5.1. Assumptions and dependencies 25](#_aeuh3t9jo62v)

[5.1.1. Assumptions 25](#_bb9gd3c0j8gg)

[5.1.2. Dependencies 25](#_bpzu8oqp77nr)

[5.2. Acronyms and abbreviations 26](#_ekw3zhfq9j4x)

# **Introduction**

## **Purpose**

The purpose of this system is to provide an accessibility-focused navigation and information platform for campus staff and students. It aims to facilitate convenient access to real-time information about campus facilities, events, and individual class schedules, thereby enhancing the campus experience and addressing the needs of users with mobility challenges.

## **Scope**

This system covers querying detailed information and real-time statuses of campus facilities, interactive campus map navigation, display of campus events and personal class schedules, as well as notification delivery. It targets all campus community members, with particular attention to users requiring accessibility accommodations.

## **Product overview**

### **Product perspective**

The system acts as a complementary digital service within the campus ecosystem, integrating with existing campus facilities management and event calendar systems. It provides a unified mobile interface supporting accessible navigation and real-time information updates.

### **Product functions**

* Query detailed information and real-time status of campus facilities such as parking lots, study rooms, and restrooms.
* Browse campus maps with clickable locations linked to Google Maps for navigation.
* View detailed information on campus events including date, time, and location.
* Access personal class schedules for enrolled students.
* Receive push notifications about facility maintenance, event changes, and class schedule updates.
* View latest news and announcements issued by the university.
* Allow users to customize notification preferences based on urgency and type

### **User characteristics**

The primary users include campus staff and students with basic smartphone proficiency, including individuals with disabilities or special accessibility requirements.

### **Limitations**

The system currently does not support offline map navigation. Real-time data depends on synchronization with campus management systems. An internet connection is required for full functionality.

## 

## **Definitions**

* **Accessibility Navigation**: Navigation designed specifically to accommodate users with mobility challenges.
* **Facility Status**: Real-time availability and condition of campus facilities.
* **Notification System**: A module that pushes important campus messages and alerts to users.
* **Push Notifications:** Automated alerts sent to user devices to inform them of important updates.
* **Kano Model:** A model to classify requirements into Must-be, Performance, and Excitement categories based on user satisfaction.

# **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.

[3] Google Maps Platform, *Google Maps API Documentation*. [Online]. Available: https://developers.google.com/maps/documentation. [Accessed: May 20, 2025].

[4] Google Forms, *Google Forms Online Survey Tool*. [Online]. Available:<https://forms.gle/HVwtyEFH2FwHaKaq9>. [Accessed: May 18, 2025].

[5] Figma, *Figma Design and Prototyping Tool*. [Online]. Available: [https://figma.com](https://www.figma.com/design/ASSktSIs8ZKbx6i9p4nl4b/SRE?node-id=1-2&t=5XKNnyQmCfEHy0SE-1). [Accessed: May 15, 2025].

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

[7] General Data Protection Regulation (GDPR), European Union, 2016. [Online]. Available:<https://gdpr.eu>. [Accessed: May 2025].

[8] Interviews and prototyping feedback sessions with MMU academic and administrative staff, May 2025, personal communications.

# 

# **Requirement**

## **Functions**

* Users shall be able to query detailed information and real-time status of campus facilities (e.g., parking availability, restrooms).
* Users shall be able to browse the campus map and click on locations to open Google Maps navigation.
* The system shall display comprehensive campus event information, including date, time, location, and details.
* Students shall be able to view their individual class schedules.
* The system shall deliver notifications regarding facility maintenance, event updates, and class schedule changes.
* Users shall be able to access the latest campus news and announcements.
* Users shall be able to customize notification preferences by type and urgency.

### **Requirements Prioritization and Classification**

#### Basic Needs

* + Navigation to campus facilities and event locations
  + Accurate, accessible maps and route guidance (including for disabled users)
  + Event integration for knowing where and when things happen
  + Clear signage (digital or physical) and basic interface usability
  + Prototyping confirmed users rely heavily on clear and accurate visual cues and map data; any inconsistencies in the prototype maps were immediately noted as confusing.

#### Performance Needs

* + Real-time updates on event changes, facility availability, and maintenance
  + Automated syncing with university systems (timetables, calendars, etc.)
  + Personalized scheduling for users (e.g., students, staff)
  + Priority reminders and alerts based on user context
  + Strongly improve satisfaction and productivity, validated by prototype interaction where users appreciated instant event updates and syncing features that reduced manual workload.
  + The prototype’s demonstration of real-time data highlighted the critical role of timely and accurate information in decision-making for both staff and students.

#### Excitement Needs

* + Indoor navigation inside complex buildings
  + AR/voice-assisted guidance
  + Customizable notification filters (urgency, type, etc.)
  + Smart suggestions: “You have 15 mins before your next event nearby”
  + Prototyping feedback emphasized that customizable notifications significantly enhance user experience by reducing alert fatigue while keeping users informed—this feature was a standout in prototype testing.
  + Interactive elements like filtering event types and setting notification urgency levels were enthusiastically received.

#### Indifferent Needs

* + General campus news unrelated to navigation/events
  + Basic reporting tools (unless directly tied to navigation problems)
  + These were ranked lower in importance during both interviews and prototype evaluation, where users focused more on actionable, time-sensitive information rather than general news.

#### **Reverse Needs**

* + Too many or irrelevant notifications (alert fatigue)
  + Invasive tracking without clear consent
  + Overcomplicated setup just to access maps/events
  + Prototyping revealed that too many notifications without filtering caused annoyance; users explicitly requested control over notification volume and type to avoid fatigue.
  + Privacy concerns around tracking were echoed, with participants urging transparent policies and opt-in consent mechanisms.

## **Performance requirements**

* The system shall respond to user queries within 3 seconds
* Notifications shall be delivered with a maximum delay of 1 minute
* System availability shall be at least 99.5% during operating hours.
* The app shall support at least 1000 concurrent users without degradation.

## **Usability requirements**

* The user interface shall be clean, intuitive, and optimized for mobile devices.
* Accessibility features such as high contrast mode and screen reader compatibility shall be supported.

## **Interface requirements**

* The system shall integrate with the campus facilities management database via a RESTful API.
* The system shall integrate with the campus event management system via a RESTful API.
* The system shall use the Google Maps API for navigation features.
* Data transmission between the app and backend services shall be secured using HTTPS and encryption.

## **Logical database requirements**

* The system shall maintain data tables for campus facilities, events, class schedules, and notifications.
* Relationships between these entities shall be logically defined to support efficient queries.

### Entity Relationship Diagram

#### *Diagram 3.1 Entity Relationship Diagram*

The Campus Accessibility Navigation System ERD outlines a user-centered design with three main roles: Student, Visitor, and Admin, all inheriting from a general User entity. Students and Visitors can plan routes, view events, and receive notifications, while Students also access class schedules. Admins manage events, update facilities, and ensure data integrity. Supporting entities include Route, which calculates and suggests paths; Event, which stores event details and accessibility impact; Facilities, which track type, status, and quantity; and Notification, which handles system messages. This structure supports efficient navigation, event awareness, and accessibility management on campus.

## **Design constraints**

* The system depends on real-time synchronization with existing campus management systems.
* The system shall be compatible with Android and iOS mobile platforms.
* The system must comply with relevant data privacy regulations (PDPA, GDPR).

## **Software system attributes**

* **Reliability**

The system shall ensure accurate and timely delivery of information and notifications.

* **Security**

The system shall protect user data privacy and prevent unauthorized access.

* **Maintainability**

The system shall be modularly designed to facilitate future updates and maintenance.

* **Privacy**

Ensure compliance with data protection laws, implement consent management and data anonymization.

* **Scalability**

Support growing number of users and data without performance loss.

## **Supporting information**

* API documentation for campus management systems integration.
* Google Maps API developer documentation

## **Behavioural Models**

### Use Case Diagram

#### *Diagram 3.2 Use Case Diagram*

### State Chart Diagram

|  |
| --- |
| *Diagram 3.3 Student Interaction State Diagram* |
| *Diagram 3.4 MMU Staff /Admin Interaction State Diagram* |
| *Diagram 3.5 System Notification State Diagram* |
| *Diagram 3.6 System Notification State Diagram* |
| **Data Synchronization State Diagram** *Diagram 3.7 Data Synchronization State Diagram* |

### Activity Diagram

|  |
| --- |
| **User Journey - Requesting and Navigating a Route (with Parking and Facilities Availability)**  This diagram will reflect the user flow, including parking and facilities availability, which directly affects the route planning.   *Diagram 3.8 User journey activity diagram* |
| **Admin Workflow - Updating Campus Data (Events, Facilities)**  This diagram will describe the administrator's process of updating the campus event and facility data in the system. *Diagram 3.9 admin workflow activity diagram* |
| **Real-Time Updates and Notifications**  This diagram represents the process of the system sending real-time updates to the users regarding campus events, maintenance updates, etc   *Diagram 3.10 Real time update and notification activity diagram* |
| **User Exiting the Navigation System**  This diagram describes what happens when the user exits the navigation system.   *Diagram 3.11 User exiting the navigation system activity diagram* |
| **System Failure/Error Handling**  This diagram outlines the process when the system encounters an error, such as failing to load campus data or calculate a route. *Diagram 3.12 system failure activity diagram* |

# 

# **Verification**

## Stakeholders and Verification Summary

|  |  |  |
| --- | --- | --- |
| **Stakeholders** | **Verification Method** | **Verification Summary** |
| Students | User acceptance testing (UAT), surveys | Verified the navigation system improves accessibility and ease of locating campus facilities. Students found the map integration and notification features helpful and intuitive. |
| MMU staff/admins | Functional testing, interviews | Confirmed system supports effective management of events and facilities. Real-time synchronization with campus databases and notification delivery was accurate and timely. |
| Software Developers (Project Team) | Unit testing, integration testing, performance testing, security testing | Ensured all modules function correctly and cohesively. Performance testing met the response time (<3s) and availability (≥99.5%) requirements under peak loads. API integrations were stable. Security testing verified encrypted data transmission and compliance with PDPA and GDPR. |

### **4.1.1 Students Verification**

**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 facility status query and display | User acceptance testing | Students can successfully query real-time facility info | Pass/Fail |
| ST-UAT-02 | Verify navigation to campus locations | User acceptance testing | Clicking map locations opens correct Google Maps navigation | Pass/Fail |
| ST-UAT-03 | Verify access to personal class schedules | User acceptance testing | Students can view their individual schedules correctly | Pass/Fail |
| ST-UAT-04 | Verify notification reception and customization | Survey and Practical Test | Students receive push notifications and can customize alert settings | Pass/Fail |
| ST-UAT-05 | Verify accessibility features (e.g., contrast, screen reader) | Accessibility testing by sample students | UI is usable and accessible by students with mobility or visual challenges | Pass/Fail |

### 4.1.2 MMU Staff / Admin Verification

**Verification Objective:** Ensure system supports staff/admin operations, event and facility management, and data synchronization.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Description | Method | Expected Result | Status |
| AD-FUNC-01 | Verify event creation, update, and deletion | Functional testing | Admins can successfully manage campus events | Pass/Fail |
| AD-FUNC-02 | Verify facility data update and status sync | Functional testing | Facility status updates correctly reflect in the system | Pass/Fail |
| AD-PERF-01 | Verify real-time synchronization with campus databases | Integration testing | Campus event and facility data syncs within 1 minute | Pass/Fail |
| AD-NOTIF-01 | Verify notification triggers for event/facility changes | Functional testing | Notifications are triggered and sent correctly to users | Pass/Fail |
| AD-SEC-01 | Verify admin access control and data security | Security testing | Admin data access is secure and authorized | Pass/Fail |

# 

### 

### 4.1.3 Software Developers (Project Team) Verification

**Verification Objective:** Confirm system modules, integration, performance, and security requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Description | Method | Expected Result | Status |
| DEV-UNIT-01 | Unit test each software module | Unit testing | Each module passes all functional unit tests | Pass/Fail |
| DEV-INT-01 | Verify integration between modules and APIs | Integration testing | Data flows correctly between system components and external APIs | Pass/Fail |
| DEV-PERF-01 | Load test system with 1000 concurrent users | Performance testing | Response times remain <3 seconds; availability ≥99.5% | Pass/Fail |
| DEV-SEC-01 | Perform security vulnerability assessments | Security testing | Data transmission is encrypted; no critical vulnerabilities found | Pass/Fail |
| DEV-ERR-01 | Test error handling and recovery mechanisms | System testing | System recovers gracefully from simulated failures and shows proper error messages | Pass/Fail |

# 

# 

## 

## 4.2 Stakeholders and Verification Summary

### **Functional Testing**

Ensured all features (facility status query, event updates, class schedule access, notifications) operate as intended with no critical defects.

### **Performance Testing**

Measured response times and concurrent user handling; system met all performance benchmarks.

### **Usability Testing**

Conducted sessions with diverse users including those with disabilities; feedback informed UI refinements.

### **Security Verification**

Conducted vulnerability assessments and encryption validation; ensured compliance with relevant data protection laws.

### **Interface Testing**

Verified RESTful API communication with external campus systems and Google Maps integration.

### **Error Handling Testing**

Simulated failures to confirm graceful error messages and system recovery mechanisms.

# 

# **Appendices**

## **Assumptions and dependencies**

### Assumptions

* Users will have consistent and reliable internet connectivity to access real-time data and notifications.
* Campus staff and students possess a basic level of proficiency in using smartphone applications.  
  The Google Maps API and other third-party services used for navigation and mapping will remain available and supported throughout the system’s life cycle.
* Integration points with existing campus management systems (facilities and event databases) will be stable and accessible via APIs.
* Users will consent to any data collection practices in compliance with relevant data protection regulations.

### Dependencies

* The system depends on continuous synchronization with campus facilities and event management systems for up-to-date information.
* Real-time notification delivery relies on mobile platform push notification services (e.g., Android Firebase Cloud Messaging, Apple Push Notification Service).
* Compliance with applicable data privacy laws such as the Personal Data Protection Act (PDPA) and General Data Protection Regulation (GDPR) is mandatory.
* The system’s performance and availability may be affected by the campus network infrastructure and third-party API service availability.

## **Acronyms and abbreviations**

* API: Application Programming Interface
* GDPR: General Data Protection Regulation
* PDPA: Personal Data Protection Act
* WCAG: Web Content Accessibility Guidelines