

**CSE6224 SOFTWARE REQUIREMENTS ENGINEERING**

**PROJECT PART 1**

By TT2L Group F

Lecturer: Nur Haifa binti Mohd Fathil

|  |  |
| --- | --- |
| **Student Name** | **Student ID** |
| Chia Kai Jun | 1211111053 |
| Wee Jia Sheen | 1211110222 |
| Yap Wei Jian | 243UC246NA |
| Tang Zhi Qian | 243UC246NP |

# Table of Content

[Table of Content 2](#_Toc201466731)

[1.0 Introduction 4](#_Toc201466732)

[1.1 Purpose 4](#_Toc201466733)

[1.2 Scope 4](#_Toc201466734)

[1.3 Product Overview 5](#_Toc201466735)

[1.3.1 Product Perspective 5](#_Toc201466736)

[1.3.2 Product Function 11](#_Toc201466737)

[1.3.3 User Characteristics 16](#_Toc201466738)

[1.3.4 Limitations 17](#_Toc201466739)

[2.0 References 19](#_Toc201466740)

[3.0 Requirements 20](#_Toc201466741)

[3.1 Function 20](#_Toc201466742)

[3.2 Performance Requirements 69](#_Toc201466743)

[3.3 Usability Requirements 70](#_Toc201466744)

[3.4 Logical Database Requirements 71](#_Toc201466745)

[3.5 Interface Requirements 74](#_Toc201466746)

[3.6 Design Constraints 75](#_Toc201466747)

[3.7 Software System Attributes 77](#_Toc201466748)

[3.8 Supporting Information 79](#_Toc201466749)

[3.8.1 Sample Input 79](#_Toc201466760)

[3.8.2 Validation Session 82](#_Toc201466761)

[3.8.3 Defect Summary 83](#_Toc201466762)

[3.8.4 Conflict Analysis 87](#_Toc201466763)

[3.8.5 Conflict Analysis Resolution 88](#_Toc201466764)

[3.8.6 Change Log 89](#_Toc201466765)

[3.8.7 Requirement Traceability Matrix 90](#_Toc201466766)

[3.8.8 Role in Requirements Validation, Negotiation & Management 91](#_Toc201466767)

[3.8.9 Version Control & Configuration Summary 91](#_Toc201466768)

[4.0 Verification 92](#_Toc201466769)

[4.1 Specified Requirements 92](#_Toc201466770)

[4.2 External Interfaces 92](#_Toc201466771)

[4.3 Functions 93](#_Toc201466772)

[4.4 Usability Requirements 93](#_Toc201466773)

[4.5 Performance Requirements 94](#_Toc201466774)

[4.6 Logical Database Requirements 94](#_Toc201466775)

[4.7 Design Constraints 95](#_Toc201466776)

[4.8 Standards Compliance 95](#_Toc201466777)

[4.9 Software System Attributes 96](#_Toc201466778)

[5.0 Appendices 97](#_Toc201466779)

[5.1 Assumptions and Dependencies 97](#_Toc201466780)

[5.2 Acronyms and Abbreviations 98](#_Toc201466781)

[5.3 Definition 99](#_Toc201466782)

# Introduction

## 1.1 Purpose

This project is to develop a ride-sharing application that helps reduce parking congestion on campus by allowing students, faculty, and staff to easily coordinate carpools. It aims to promote more efficient, convenient platforms and ensure only verified university members can access the platform through digital ID verification.

## 1.2 Scope

This project will focus on building a ride-sharing application for members of the university community. The system will:

* Allow user to create and join carpools
* Secure digital ID verification to restrict access to verified students, faculty, and staff
* Integrate with the campus parking management system to show real-time parking availability

## 1.3 Product Overview

### 1.3.1 Product Perspective

*1.3.1.1 System Interface*

The platform will incorporate several systems to carry out its functionalities. The system interfaces include:

University Parking System:

1. Establish connection to the university’s parking system to allow for real-time checking and update users about parking availability.

University Database System:

1. Establish a connection to the university’s current database to access and update user’s profile and records of carpool
2. Make sure data remains consistent and trustworthy between the university student information system (SIS) and campus ride-sharing platform.

*1.3.1.2 User Interface*

The following table summarizes the list of graphical user interface (GUI) requirements specified for campus ride-sharing platform.

**Table 1.3.1.2 Graphical User Interface (GUI) Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Authors |
| REQ\_GUI1 | Campus ride-sharing platform GUI shall use a color scheme that is compliant with university’s official colors. For example, the university official colors are blue (RGB Hex code: #0D44FF) and red (RGB Hex code: #CE2121), we will be mainly using these for the interface element. Ensure color contrast meets WCAG 2.1 standards for accessibility (e.g., contrast ratio ≥ 4.5:1 for normal text). | High | Chia Kai Jun |
| REQ\_GUI2 | The dashboard shall present a layout that displays upcoming rides, active carpools, and real-time parking availability, ensuring key actions are accessible within **2 clicks** and the user drop-off rate remains below **10%** during navigation. | High | Chia Kai Jun |
| REQ\_GUI3 | The platform interface shall support responsive design across devices, with breakpoints defined as follows: **mobile (≤ 480 px), tablet (481–1024 px), and desktop (≥ 1025 px)**. | High | Chia Kai Jun |
| REQ\_GUI4 | The navigation bar shall remain fixed on top of all pages, with clearly labeled tabs | Medium | Chia Kai Jun |

*1.3.1.3 Hardware Interface*

The following table summarizes the list of hardware requirements specified for campus ride-sharing platform.

**Table 1.3.1.3 Hardware Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Authors |
| REQ\_H1 | The backend server shall be hosted on a cloud-based platform with at least 4 vCPUs, 16 GB RAM, and 100 GB SSD storage. | High | Chia Kai Jun |
| REQ\_H2 | The platform must be compatible with iOS and Android mobile devices, specifically supporting **iOS versions 14 to 17** and **Android versions 11 to 14**. | High | Chia Kai Jun |
| REQ\_H3 | All physical interaction devices used to access the platform, such as keyboards, mice, and touch screens, must be supported to ensure that all features function properly. | High | Chia Kai Jun |

*1.3.1.4 Software Interfaces*

The following table summarizes the list of software requirements specified for campus ride-sharing platform.

**Table 1.3.1.4 Software Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Authors |
| REQ\_S1 | The system shall be developed using a web framework such as Django with PostgreSQL as the database. | High | Chia Kai Jun |
| REQ\_S2 | The mobile application shall be built using React Native to ensure cross-platform compatibility | High | Chia Kai Jun |
| REQ\_S3 | The platform must integrate with Single Sign-On (SSO) authentication for secure login and data fetching. | High | Chia Kai Jun |

*1.3.1.5 Communication Interface*

The following table summarizes the list of communication requirements specified for campus ride-sharing platform.

**Table 1.3.1.5 Communication Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Authors |
| REQ\_C1 | The system shall support real-time communication between ride offers and ride seekers via in-app messaging. | High | Chia Kai Jun |
| REQ\_C2 | The system shall ensure secure communication between client apps and backend services using HTTPS. | High | Chia Kai Jun |

*1.3.1.6 Memory Constraints*

* Server hosting the platform should have a minimum of 16GB RAM.
* A minimum of 8GB RAM should be dedicated to caching processes.

*1.3.1.7 Operation*

Automated monitoring systems must be implemented to track system performance, including server load, response time, and transaction volumes, with real-time alerts sent to system administrators for immediate actions taken in case of any issues.

To avoid service disruption, platform maintenance should be scheduled during off-peak hours and communicated to users at least 48 hours in advance through in-app announcements.

Periodic data integrity checks should be undertaken to ensure consistency across all integrated systems. Any anomalies shall be logged and reported for the admin to review.

Daily automated backups of the system should be performed to secure data integrity and availability.

*1.3.1.8 Site Adaptation Requirements*

The platform should change its content and layout to match local customs and standards, like date formats, currency, and measurement units, to make sure it's easy and relevant for users in different regions.

*1.3.1.9 Interfaces with Service*

Utilizing cloud services for storage allocation, processing power, and network administration can ensure scalability, dependability, and performance.

Integration with a Customer Relationship Management (CRM) system to manage rides, track interactions, and streamline communication.

### 1.3.2 Product Function

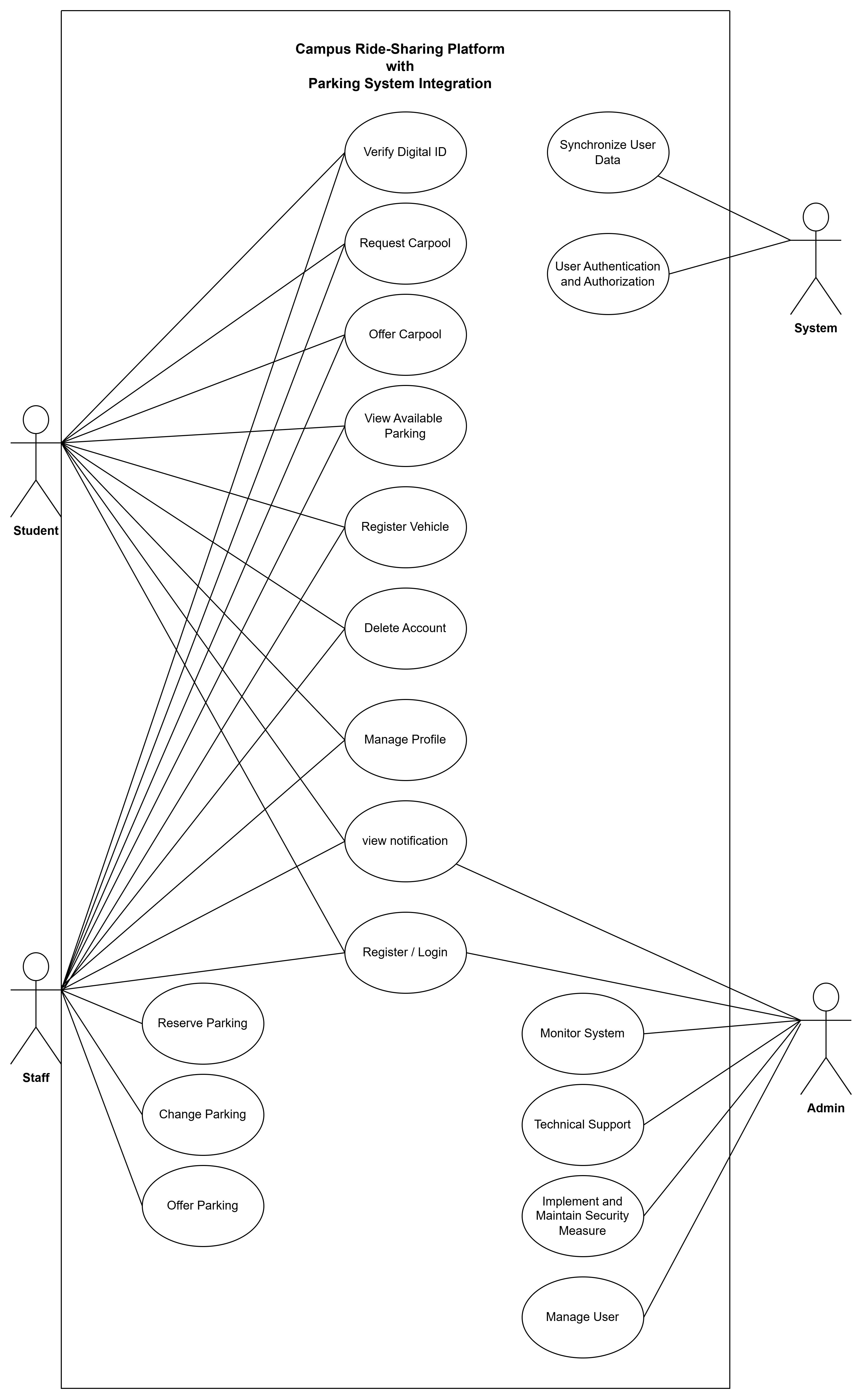


Figure 1.3.2 Use Case Diagram for Campus Ride-Sharing Platform

**Use Case Specification**

|  |  |  |
| --- | --- | --- |
| **Actor** | **Use Cases** | **Specification** |
| Student & Staff & Admin | View Notification | **Description**: Notify users with timely reminders to update their profile for uninterrupted access  **Precondition**: User has logged into the system  **Postcondition**: User is informed of incomplete profile and prompted to update it |
| Student & Staff | Login | **Description**: Allow student/staff login to their account using existed ID and passwords in the university’s database  **Precondition**: Users have access the registration page  **Postcondition**: Users inputs are valid and able to log in |
| Verify Digital ID | **Description**: Allow students and staff to verify their identification to access the platform’s function  **Precondition**: User is logged into the system and has access to their university-issued digital ID  **Postcondition**: User’s identity is authenticated and verified, granting access to protected platform features |
| Request Carpool | **Description**: Students and staff shall be able to request for joining carpools with available seats  **Precondition**: Users are able to navigate to request page and see available carpool  **Postcondition**: User joins the requested carpool |
| Offer Carpool | **Description**: Students and staff shall be able to offer carpools to other users and decide whether to accept or decline the request  **Precondition**: User is able to navigate to offer carpool page and has verified their ID  **Postcondition**: Carpool offer is successfully posted and available to other users |
| Manage Profile | **Description**: A Student/Staff user manages their profile. The system authenticates the user, processes profile updates, saves valid changes to the database, and provides success or error notification, or redirects to login.  **Precondition**: Student/Staff user has an existing account.  **Postcondition**: Student/Staff profile is updated in the database, or an error/redirection occurs, with user notification. |
| View Available Parking | **Description**: Students and staff able to view available parking in the university in real time  **Precondition**: User is logged into their account and has verified their ID  **Postcondition**: User can view the available parkings |
| Register Vehicle | **Description:** Students and staff shall be able to register their vehicle to the system so that they can offer carpools and other functions that required owning a vehicle  **Precondition:** User have not registered a vehicle before and have access to the vehicle registration page **Postcondition:** User’s vehicle is registered and eligible to offer carpool. Staff can reserve parking |
| Staff | Delete Account | **Description**: A Student/Staff user deletes their account by confirming deletion in the Application System. The system sends a DELETE request to the Server, which removes the user's data from the Database and confirms deletion to the user.  **Precondition**: Student/Staff user is logged in with an existing account.  **Postcondition**: User's account and data are permanently deleted, and the user receives a confirmation. |
| Reserve Parking | **Description**: Staff shall have priority to reserve available parking spaces beforehand, with a limit of one spot per staff member.  **Precondition**: Staff is logged into the system and has verified digital ID.  **Postcondition**: A parking space is reserved and marked unavailable for others. |
| Change Parking | **Description**: Staff have the right to request change the parking before a day.  **Precondition**: Staff are authenticated and have a reserved parking space.  **Postcondition**: A staff reserved parking of staff is swapped with another chosen parking. |
| Offer Parking | **Description**: Staff can release or offer their parking spot to others if they don’t need it.  **Precondition**: Staff have an active parking reservation.  **Postcondition**: Parking is made available for others to request. |
| Admin | Login | **Description**: The admin logs into the Campus Ride-Sharing Application System by providing credentials, which are verified by the server against the database. Successful login displays the admin dashboard; failed login prompts an error message.  **Precondition**: The Campus Ride-Sharing Application System is operational.  **Postcondition**: Admin is either successfully logged in to the dashboard or presented with a login failed message, with all login attempts logged. |
| Monitor System | **Description**: Admin shall continuously monitor system performance, uptime, and logs.  **Precondition**: Admin is logged into the backend system.  **Postcondition**: System metrics are logged and anomalies flagged for review. |
| Technical Support | **Description**: Admin provides support for users experiencing issues with platform functionality.  **Precondition**: Admin has access to system and user support tools.  **Postcondition**: Reported issues are addressed, logged, and resolved or escalated. |
| Implement and Maintain Security Measure | **Description**: Admin ensures the system is secure through updates, encryption, and access control.  **Precondition**: Admin privileges verified, and access granted.  **Postcondition**: Security configurations are updated and active, logs are stored. |
|  | Manage User | **Description**: An Admin manages users via the Admin Interface, viewing, editing roles, or deactivating accounts. The system interacts with the Database to perform these actions and provides feedback.  **Precondition**: Admin is logged in with user management permissions.  **Postcondition**: User accounts are viewed, modified, or managed in the database, and the Admin is notified. |
| Internal System | Synchronize User Data | **Description**: Internal system synchronizes user data across all modules and services.  **Precondition**: User data is changed or actions are performed.  **Postcondition**: All modules reflect the updated data consistently. |
| User Authentication and Authorization | **Description**: Internal system verifies user credentials and grants access based on roles.  **Precondition**: User submits valid login credentials.  **Postcondition**: User is authenticated and redirected to their role-specific dashboard. |

### 1.3.3 User Characteristics

The following table shows the expected level of knowledge for each role.

**Table 1.3.3 User Characteristics**

|  |  |  |
| --- | --- | --- |
| Role | Description | Require Knowledge |
| Student | Students may use the platform to request or offer carpools. | Basic understanding on how to use campus ride-sharing platform.  Aware of privacy settings and protection principles. |
| Staff | Staff may use the platform to request, offer carpools, and reserve parking. | Basic understanding on how to use campus ride-sharing platform. |
| Admin | Admins need to ensure platform operates smoothly. They will handle technical maintenance, updates, and security protocols. | Ability to troubleshoot system and technical support.  Advance knowledge of network and database management.  Have skills in cybersecurity and data protection, |

### 1.3.4 Limitations

This section provides descriptions of limitations that could affect the design and functions of Campus Ride-sharing Platform.

*1.3.4.1 Regulatory Requirements and Policies*

**National Institute of Standards and Technology (NIST)**

The system must comply with campus IT security policies related to authentication and adhere to the standards and recommendations established by the National Institute of Standards and Technology (NIST). Therefore, this platform must integrate with existing campus identity providers and follow strict control standards.

*1.3.4.2 Performance and Scalability*

The Campus Ride-sharing Platform must handle peak usage without lag or crashes. The system shall support a minimum of 500 concurrent users during peak university hours (8-9 AM, 5-6 PM). To achieve this, it requires scalable backend architecture and load balancing.

*1.3.4.3 Data Privacy and Security Regulations*

The platform must implement robust encryption mechanisms to protect user information from unauthorized access.

The system must comply with privacy laws (e.g.,PDPA) concerning the collection, storage, and sharing of personal and location data. The platform should only collect personal data that is used only for the purposes for the platform’s operation and implement appropriate security measures to protect personal data. This will affect design decisions for authentication and encryption.

The platform must also comply with WCAG 2.1 accessibility standards to ensure usability for individuals with disabilities, including screen reader compatibility, keyboard navigation, and sufficient color contrast.

*1.3.4.4 Limitations from Other Systems*

**Third-Party Dependency**

The Campus Ride-sharing Platform is dependent on third-party systems for data and features could result in constraints due to their availability, performance and policy changes.

* The uptime of third-party services may impact the platform’s functionality
* Performance of the platform may impacted by speed of third-party systems
* The changes of policies of the third-party APIs provider may causes the platform to be modified in response to the adjustments

# 2.0 References

<https://pages.nist.gov/800-63-4/>

<https://www.iso.org/standard/35733.html>

<https://educationmalaysia.gov.my/privacy-policy/>

<https://advisera.com/27001academy/what-is-iso-27001/>

<https://swagger.io/specification/>

<https://clym.io/regulations/malaysia-personal-data-protection-act-pdpa>

<https://www.digitalguardian.com/blog/what-role-based-access-control-rbac-examples-benefits-and-more>

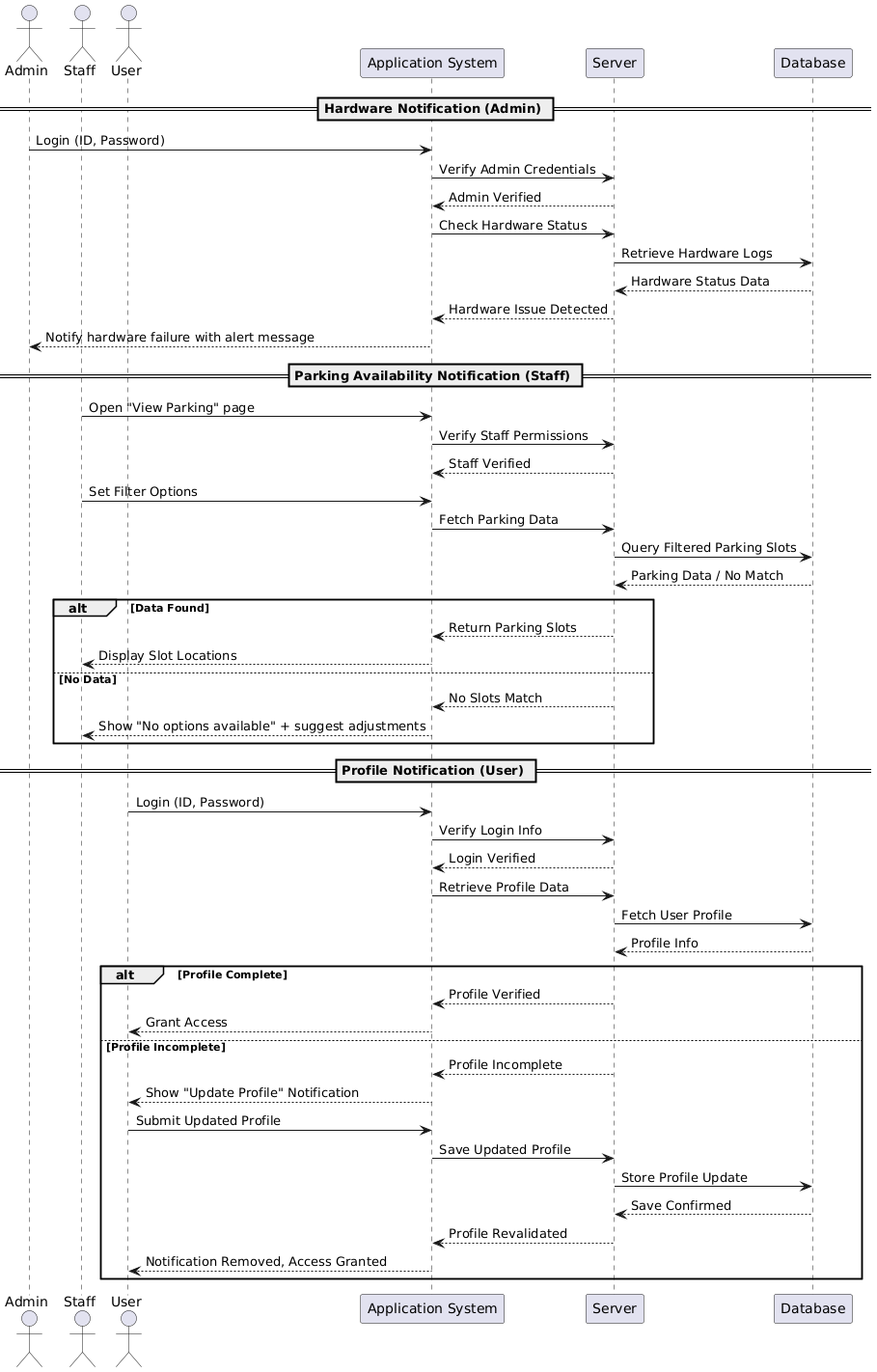
https://www.w3.org/TR/WCAG21/

# 3.0 Requirements

## 3.1 Function

3.1.1 Student/ Staff / Admin View Notification

**Sequence Diagram**

 Figure 3.1.1 Sequence Diagram for Student / Staff / Admin View Notification

1. **Authentication and Verification**
   1. The system validates whether the user is logged in with a valid admin, staff, or user account.
   2. **The system checks whether the respective role's permissions are sufficient before allowing access to the related features (e.g., admin hardware monitor, staff parking view, user profile update).**
2. Responses to Abnormalities
   1. **Hardware Failure (Admin Role)**  
      **Issue**: The servers, storage devices, or hardware components hosting the system fail to function normally.  
      **Response**:
      * Regularly monitor hardware conditions.
      * Automatically notify admins upon detecting hardware failure.
      * Display user-facing error messages indicating temporary unavailability.
   2. No Parking Slots Available (Staff Role)

Issue: No available parking slots based on the user’s applied filters.

Response:

* + - * + Show notification stating “No options available.”
        + Suggest users adjust their filters for broader results.
        + Ensure real-time parking slot updates are fetched from the system.
  1. Incomplete User Profile (User Role)

Issue: Incomplete or outdated profile information can restrict access to the system.

Response:

* Prompt the user to complete or update their profile.
* Use regular notifications to remind users about outdated information.

1. Effect of parameters
   1. Profile Completeness

Problem: Incomplete or outdated profile information can restrict access to the system.

Solution:

* + - Implement mandatory checks for profile completeness.
    - Send reminders and visual alerts to users for updating missing or outdated data.

1. Relationship of outputs to inputs
   1. Input / Output Sequence – View Available Parking (Staff Role)
      1. User opens the “View Parking” page.
      2. The system checks the user’s status and permissions.
      3. User applies filters based on preferred options (e.g., location, availability).
      4. The system fetches and displays real-time parking slot data.
      5. If filters match data, available parking locations are shown.
      6. If no data matches, a message "No options available" is displayed with filter adjustment suggestions.
   2. Input / Output Sequence – Hardware Monitoring (Admin Role)
      1. Admin accesses the hardware monitoring dashboard.
      2. System continuously monitors server and storage health.
      3. On detecting an anomaly or failure, the system logs the event.
      4. Admin receives automated notifications.
      5. A user-friendly error page is shown to users during disruptions.
   3. Input / Output Sequence – Profile Update (User Role)
      1. User logs into the system and accesses their profile.
      2. System checks for completeness of required profile fields.
      3. If data is missing or outdated, a notification prompts for updates.
      4. User completes the update and submits.
      5. System validates the input and updates the profile.
      6. Confirmation message is displayed once successful.

3.1.2 Student / Staff Login to Account

**Sequence Diagram**

A diagram of a login

AI-generated content may be incorrect. Figure 3.1.2 Sequence Diagram for Student / Staff Login

1. Validity Checks on Inputs
   1. Ensure student identification provided exists in university student information system (SIS)
   2. Make sure password provided is validated and linked to the corresponding student ID
2. Responses to Abnormal Situations during Registration and Login
   1. Overflow

Issue: The system’s resources may get overwhelmed if there’s too many users attempting to login simultaneously.

Response: Implement measures such as server balancing or queueing mechanisms to manage high traffic and prevent overload. Inform the users about the high traffic and advise them to try again later.

* 1. Communication Error

Issue: The network infrastructure and services required for communication between users’ devices and the server hosting the login platform fail to operate correctly.

Response: Use automated monitoring tools to detect downtime and send alerts to admins when communication failure is detected. Meanwhile, also keep the users informed when this situation occurs for acknowledgement.

* 1. Hardware Failures

Issue: The servers, storage devices or hardware components hosting the system fail to function normally.

Response: Regularly monitor the conditions of hardware components and promptly replace any failing hardware. Implement automated notifications to notify admins in case of hardware failures and provide error messages to users.

1. Effect of parameters
   1. User Profile Initialization

Problem: On first successful login, the system needs to create a user profile using information from the university student information system (SIS). If it fails, the user is unable to access the system.

Solution: implement checks and fallback logic for partial profile creation

1. Relationship of outputs to inputs
   1. Input/Output Sequences (Login)
      1. Students access the login page.
      2. System displays a login form with fields for ID and password.
      3. Students enter their ID and password into the form.
      4. The system verifies the provided credentials and authenticates the student.
      5. If the credentials match the ones in the database, the system grants access to the student’s account and redirects it to their dashboard.

3.1.3 Student/Staff Verify Digital ID

**Sequence Diagram**

A diagram of a diagram

AI-generated content may be incorrect. Figure 3.1.3 Sequence Diagram for Student / Staff Verify Digital ID

1. Authentication and Verification
2. Verification from university-issued credentials, which is Student / Staff ID should be concerned
3. Establishing identity trust to ensure that the user is valid in the institution records
4. Responses to Abnormalities During Digital ID Verification
5. Concurrent Login Attempts

Issue: Large amounts of student attempting to login simultaneously.

Response: Use queuing systems to process requests.

1. System Sync Delay

Issue: The university student information system (SIS) does not update any changes recently.

Response: Display message “ID not found, please try again or contact support.”

1. Data Inconsistency Between Systems

Issue: System failed to validate ID due to missing records in university’s database.

Response: Notify the user to log the issue and route a backend ticket for data synchronization.

1. Effect of Parameter
2. Invalid or Inactive ID

Problem: A user attempts to verify their Digital ID, but fails to verify due to ID being expired, deactivated or not recognized by the university student information system (SIS).

Solution: Show message to the user to notify related to the ID about “Digital ID not active or invalid, please contact university support” and block further access.

1. Relationship of outputs to inputs
2. Input / Output Sequence (Verify Digital ID)
3. User open Campus Ride-Sharing Application System
4. System prompts for university Digital ID Verification
5. User submit their Digital ID
6. System check ID status with university student information system (SIS).
7. If Verification successful, then proceed to dashboard.
8. If Verification failed, show error message and block access.

3.1.4 Student/Staff Request Carpool

**Sequence Diagram**

A diagram of a process flow

AI-generated content may be incorrect. Figure 3.1.4 Sequence Diagram for Student/Staff Request Carpool

1. Authentication & Verification
2. The system should ensure user profile is registered and detailed, including pick-up location (most preferred), contact information and campus status, (either student or staff).
3. The system also verifies the requested time slot and destination are within the supported service range and operating hours.
4. Response to Abnormalities during Carpool Request Carpool
5. Missing Request Information

Issue: User submit a carpool request without selecting date and time.

Response: Prompt a message alert user to fill out required information such as “Please select your appropriate date and time before submitting”.

1. Unavailable Routes

Issue: Requested route was full or not operating.

Response: Suggest user to alternate time slots or add user to waiting list.

1. System Latency

Issue: System delay when user send requests to the server.

Response: Prompt message to user about “Request Time Out” to try again.

1. Effect of Parameter
2. Ride Preferences and Filters

Problem: Too many filters will result with constant unavailable carpools.

Solution: Prompt users to decompress their references to have more ride options.

1. Relationship of Outputs and Inputs
2. Input / Output Sequence (Carpool Request)
3. User navigates to the carpool request page.
4. System display form with different options
5. User fills in request form and submits it.
6. System verifies all fields and checks for available matching carpools.
7. If found matching carpool request, the confirmation will be shown for user to add schedule
8. If matching carpool request not found, a error message will pop out to notify user to try again or change preferences.

3.1.5 Student/Staff Offer Carpool

**Sequence Diagram**

A diagram of a process

AI-generated content may be incorrect. Figure 3.1.5 Sequence Diagram for Student / Staff Offer Carpool

1. Authentication and Verification
2. The system validate whether users logged in and verified before offering a carpool.
3. Also includes the vehicle details, driver eligibility and contact information in addition to campus ride-sharing rules and regulations.
4. Response to Abnormalities during Carpool Offer
5. Incomplete Driver Profile

Issue: User tries to offer a ride without providing vehicle and license information

Response: Block the submission and prompt “Please complete your vehicle registration before offering a carpool” error message.

1. Conflicting with Ride Schedules

Issue: User schedule multiple carpools with different time and date

Response: Show warning message about “You have an active carpool offer” and block the submission.

1. System Error and Delay

Issue: System fails to save the carpool offer due to technical issues.

Response: Inform user about “submission failed, please try again” error message and auto-save the submission as draft.

1. Effect of Parameter
2. Route and Time Selection

Problem: Offering a ride on non-peak hour and odd time may result in no matching riders.

Solution: Suggest about setting up routes at peak hour to establish efficiency ride.

1. Relationship of Outputs to Inputs
2. Input / Output Sequence (Offer Carpool)
3. User accesses the “Offer Carpool” page.
4. System shows Offer Carpool form with input time, destination, seat availability and vehicle details.
5. User submit offer.
6. System validates and confirm the ride offer.
7. If Carpool Offer is valid, then ride options will be listed and visible to riders.
8. If Carpool Offer is invalid, then an error message will pop out and guide user to correct input.

3.1.6 Student/Staff View Available Parking

**Sequence Diagram**

A diagram of a project

AI-generated content may be incorrect. Figure 3.1.6 Sequence Diagram for Student / Staff View Available Parking

1. Authentication and Verification
2. The system assure user logged in and verified as an active members, whether it is student or staff.
3. Vehicle should be registered in the system and limited to authorized parking zone depending on user status.
4. Response to Abnormalities during Parking View
5. Invalid User Status

Issue: User does not have parking privileges due to not register vehicle into the system.

Response: Display message “You are not eligible to access parking information, please register your vehicle.”

1. No Real-Time Data Available

Issue: Real-time parking availability cannot be read due to server issues.

Response: Inform user about message “Live parking data currently unavailable, please try again”

1. Location Access Denied

Issue: The application cannot be user’s current location due to permission denied.

Response: Prompt user to enable location access to view available parking slot.

1. Effect of Parameter
2. Filter by Parking Zone or Time

Problem: Filtering by specific parking lots or peaks hour may result in no available parking displayed in the application.

Solution: Suggest widening search range or remove filter options to provide more and better result.

1. Relationship of Outputs and Inputs
2. Input / Output Sequence (View Available Parking)
3. User opens the “View Parking” page
4. System check user’s status and permissions
5. User set filters with their preferred options
6. System fetches and display available parking slot in real-time
7. If the filter match the data, the application will show the location of available parking
8. If the filter does not match the data, a notification will display saying “no options available” and suggest for adjustment.

3.1.7 Student/Staff Register Vehicle

**Sequence Diagram**

A screenshot of a diagram

AI-generated content may be incorrect. Figure 3.1.7 Sequence Diagram for Student / Staff Register Vehicle

1. Authentication and Verification
2. The system validate whether the user is logged in with valid student / staff account.
3. Also validate if user identity is verified before the system allow them to access vehicle registration page
4. Response to Abnormalities during Vehicle Registration
5. Invalid Vehicle Information

Issue: User submits form with empty or incorrect plate number / vehicle type

Response: Prompt user to input valid vehicle details to continue

1. Vehicle Already Registered

Issue: The same vehicle is already registered under another account.

Response: Show error message about “the vehicle is already registered by another user.”

1. System Timeout & Submission Error

Issue: Vehicle Registration fails due to network issues.

Response: Inform user about fail to process registration and ask them to try again later.

1. Effect of Parameter
2. Vehicle Type and Plate Format

Problem: Unsupported vehicle types or incorrect plate formats will result in registration fail.

Solution: Provide clear format guidelines and validate inputs before submission.

1. Relationship of Outputs to Inputs
2. Input / Output Sequence (Register Vehicle)
3. User navigates to the “Register Vehicle” page.
4. System display form which requesting for vehicle information
5. User enters and submits the information
6. System validates input and check for existing records in the university student information system (SIS)
7. If vehicle has not registered yet, the vehicle registration will be successful, then the confirmation will be shown.
8. If vehicle has been registered before, then the vehicle registration will be failed, afterwards display an error message.

3.1.8 Student/Staff Delete Account

**Sequence Diagram**

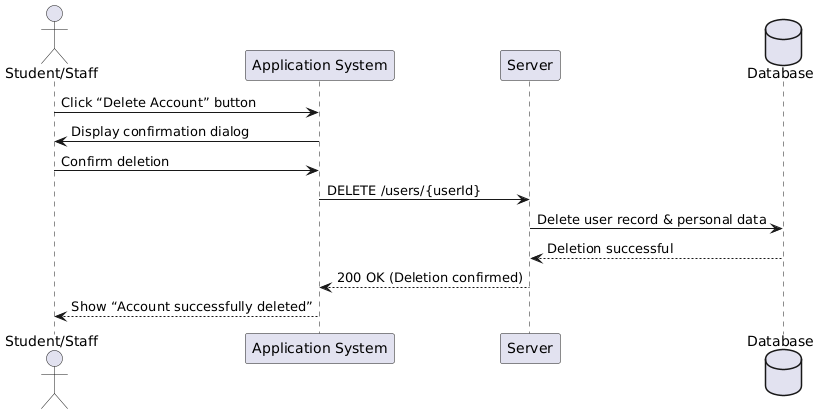
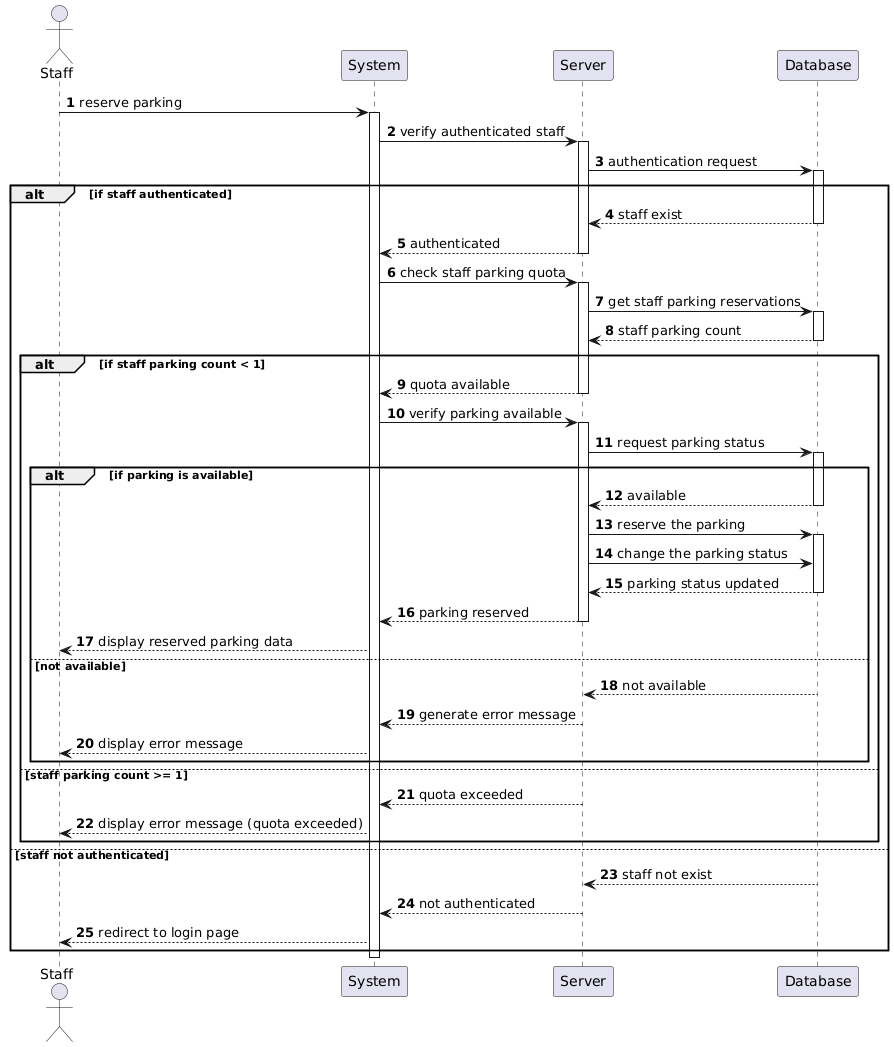


Figure 3.1.8 Sequence Diagram for Student / Staff Register Account Deletion

1. Authentication & Authorization
   1. Validate that the user is logged in with a valid student/staff session
   2. Verify that the authenticated user’s identity matches the account being deleted
   3. (If enabled) Confirm any additional identity checks (e.g. 2FA) before proceeding
2. Response to Abnormalities during Account Deletion
   1. Missing Confirmation
      * ****Issue****: User closes or cancels the confirmation dialog without confirming.
      * ****Response****: Abort deletion; return to account settings with a message “Account deletion canceled.”
   2. Unauthorized Access
      * **Issue**: User attempts deletion via direct API call without proper auth token or with expired session.
      * **Response**: Return HTTP 401 Unauthorized with “Please log in to delete your account.”
   3. Deletion Error / System Failure
      * **Issue**: Backend deletion operation fails (e.g. database timeout, constraint error).
      * **Response**: Return HTTP 500 Internal Server Error with “We couldn’t process your request—please try again later.”
3. Effect of Parameter
   1. User role policy
      * Apply soft-delete vs. hard-delete rules based on role (e.g. archive staff accounts, purge student accounts)
   2. Data retention period
      * If a retentionPeriod parameter is set, mark data as “deleted” in the UI but retain in archive for the specified duration
4. **Relationship of Outputs to Inputs (Account Deletion Flow)**
   1. User navigates to “Account Settings → Delete Account”
   2. System displays confirmation prompt (e.g. “Type DELETE to confirm”)
   3. User enters confirmation value and submits
   4. System validates authentication, identity match, and any parameters
   5. System deletes or archives data per policy
   6. System returns success (200 OK) or appropriate error code/message

3.1.9 Staff Reserve Parking

**Sequence Diagram**

Figure 3.1.9 Sequence Diagram for Staff Reserve Parking

1. Authentication
2. Ensure the staff is logged into the system.
3. Verification to Parking Status
4. Ensure the parking status is available before reserve the chosen parking.
5. Response to Abnormal Situation during Reserve Parking
6. Network Failure

Issue: When user has connectivity issues while attempting the reservation.

Response: Network connection lost. Please check your internet connection and try again.

1. Unauthorized Access

Issue: When the system detects a user tries to reserve a slot without being logged in or authorized.

Response: You must be logged in to reserve a parking slot. Please login to your account.

1. Parking Slot Already Reserved

Issue: When the system detects the user tries to reserve a parking slot that is already booked.

Response: This parking spot has already been reserved. Please choose another available slot.

1. Relationship of input to output
2. Input/Output Sequences (reserve parking)
3. Staff access the reserve parking page.
4. The system displays all the reserved and available parking slots with UI.
5. Staff choose a parking spot with available status.
6. The system verifies the chosen parking status and authenticates the staff.
7. If the parking status is available, the system will reserve the parking slot for the staff and change the parking status to reserved state.
8. After reserving the parking slot, the system displays a successful message to the staff.

3.1.10 Staff Offer Parking

**Sequence Diagram**

 Figure 3.1.9 Sequence Diagram for Offer Parking

1. Authentication
2. Ensure the staff is logged into the system.
3. Offered User Verification
4. Ensure the user who receives the offer parking exists.
5. Abnormal situation response during Offer Parking
6. Network Failure

Issue: When staff has connectivity issues while offering parking to the user.

Response: Network connection lost. Please check your internet connection and try again.

1. Unauthorized Access

Issue: When the system detects a staff trying to offer parking to a user without being logged in or authorized.

Response: You must be logged in to reserve a parking slot. Please login to your account.

1. Missing parking

Issue: When staff try to offer parking without having any reserved parking space.

Response: Offer parking failed. You must have at least one reserved parking space to offer.

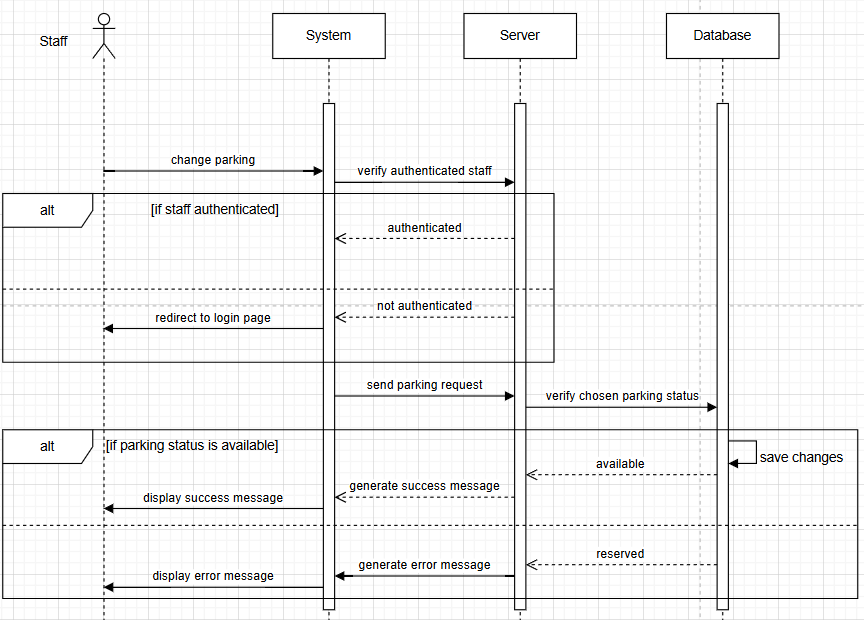
1. Relationship of input to output

a. Input/Output Sequences (Offer Parking)

1. Staff access the offer parking page.
2. System verifies the offer parking that has belonged to the staff.
3. If the offer parking belonged to staff, the system will remove the staff reserved parking and change it status to available.
4. The system displays the success message to staff.

3.1.11 Staff Change Parking

**Sequence Diagram**

Figure 3.1.11 Sequence Diagram for Change Parking

1. Authentication
2. Ensure the staff is logged into the system.
3. Verify Parking Status
4. Ensure the parking status available before request parking changes.
5. Response to abnormal situation during Change Parking
6. Parking reserved

Issue: When the system detects the chosen change parking request is reserved state.

Response: The parking is reserved. Please choose another parking spot.

1. Unauthorized Access

Issue: When the system detects a staff tries to offer parking to a user without being logged in or authorized.

Response: You must be logged in to reserve a parking slot. Please login to your account.

1. Network Failure

Issue: When staff has connectivity issues while offering parking to the user.

Response: Network connection lost. Please check your internet connection and try again.

1. Relationship of input to output
2. Staff access to request parking page.
3. The system displays all the parking with UI.
4. Staff choose a car park to request changes.
5. The system validates the parking status of chosen parking.
6. If parking spot status is available, the system changes the originally reserved parking status to available and the chosen parking status to reserved.
7. The system will deliver a successful message to staff.

3.1.12 Staff Manage Profile

**Sequence Diagram**



Figure 3.1.12 Sequence Diagram for Staff Manage Profile

1. Authentication
2. Ensure the staff is logged into the system.
3. Data Validation
4. Ensure updated profile data is validated before update profile.
5. Response Abnormal Situation during manage profile
6. Unauthorized Access

Issue: When the system detects a staff tries to offer parking to a user without being logged in or authorized.

Response: You must be logged in to reserve a parking slot. Please login to your account.

1. Network Failure

Issue: When staff has connectivity issues while offering parking to the user.

Response: Network connection lost. Please check your internet connection and try again.

1. Missing Required Information

Issue: When staff update profile data without any data that needs to be required.

Response: Please complete all required fields to update your profile.

1. Relationship input to output
2. Staff access the profile page.
3. The system displays the existing data for staff to update their profile.
4. Staff input the data to the form and click save to update profile.
5. The system validates the profile data is valid.
6. If the profile data is valid, the system updates and saves the data through the database.
7. The system displays a successful message to staff.

3.1.13 Admin Login

**Sequence Diagram**



Figure 3.1.13 Sequence Diagram of Admin Login

1. Authentication and Validation
2. The system ensures user login is authenticated as an Admin by verifying their username and password from the admin database record. Admin username and password format should be differ compared to user.
3. After successfully login, the system confirmed user’s role as Admin and enable to access administrative functionalities such as review system logs.
4. Response to Abnormalities during Login
5. Invalid Login Information

Issue: The admin enters incorrect login information (username and password)

Response: Display error message about login failed and request user to check the username and password.

1. Unauthorized Access Attempt

Issue: Non-admin user tries to access admin login page

Response: Display message about not granted access and block the authorization of admin system login

1. Server Unavailable

Issue: The authentication system is not reachable

Response: Display message about unable to connect server and ask to try again.

1. Effect of Parameter
2. Session Timeout

Problem: Admin logged in the system but will be inactive for certain time after inactivity.

Solution: Prompt warning timeout message and redirect to login page to request admin log in again

1. Relationship of Outputs and Inputs
2. Input / Output Sequence (Admin Login)
3. Admin navigates to the “Admin Login” page
4. Admin inputs username and password
5. System validates the username and password from the database
6. If username and password matches with admin permission, then login will be successful. Then redirect to admin dashboard page.
7. If username and password does not match, then system will display error message.

3.1.14 Users Manage Profile

**Sequence Diagram**

A screenshot of a computer screen

AI-generated content may be incorrect.Figure 3.1.14 Sequence Diagram for User Manage Profile

1. Authentication and Verification
2. The systems will confirm the users is authenticated and verified.
3. Once the users is authenticated verified, the profile edit options will be enable
4. Response to Abnormalities during Profile Management
5. Invalid Input Format

Issue: User enter their profile information in the wrong format

Response: Prompt error message for user to input valid profile information

1. Update Conflict

Issue: Server delay during profile update.

Response: Alert user about fail to update the profile and offer to retry.

1. Unauthorized Access Attempt

Issue: User attempt to change restricted fields

Response: Deny the action in addition with warning message about no permission to do the action.

1. Effect of Parameter
2. Profile Completeness

Problem: Incomplete or outdated profile can result in prohibited to access the system

Solution: Prompt users to frequently update their profile with regular notification

1. Relationship of Outputs to Inputs
2. Input / Output Sequence (Manage Profile)
3. User access “Edit Profile” page.
4. System fetches and display current profile information
5. User update profile and submit changes
6. System validates inputs and update records
7. If the information is valid, then the profile will be updated in the system with confirmation.
8. If the information is invalid, then an error message will be prompt and request the user to fix the input.

3.1.15 Admin Manage Users

**Sequence Diagram**



Figure 3.1.15 Sequence Diagram for Admin Manage Users

1. Authentication
2. Ensure the admin is logged into the system.
3. Response Abnormal Situation during monitor system
   1. **Communication Error**  
      Issue: Admin’s device fails to establish a connection with the server while retrieving or updating user data.  
      Solution: Display a “Connection Lost” message. Retry the operation or allow the admin to save changes temporarily for syncing later. Notify IT support via alerts.
   2. **Permission Error**  
      Issue: Non-admin user attempts to access admin management functions.

Solution: Deny access with a clear error message and log the unauthorized attempt for security auditing.

1. Effect of Parameters
2. **User Role Validity**

**Problem**: If the user's role is incorrectly assigned or missing (e.g., a standard user marked as admin), unauthorized access to sensitive features may occur or legitimate access could be denied.

**Solution**: Ensure proper role assignment during account creation and include role validation checks before granting access to admin functions.

1. Relationship of output to input
2. Input / Output Sequence
3. Admin logs in and accesses the user management panel.
4. System displays a list of registered users (students/staff), with filter and search options.
5. Admin selects a user to update, deactivate, or delete.
6. Admin performs the desired action (e.g., updates user status to "inactive").
7. System verifies admin’s permission, then applies changes to the database.
8. A confirmation message is displayed and the change is logged for audit purposes.
9. Updated user list is shown reflecting the changes.

3.1.16 Admin Monitor System

**Sequence Diagram**



Figure 3.1.16 Sequence Diagram for Admin Monitor System

1. Authentication
2. Ensure the admin is logged into the system.
3. Response Abnormal Situation during monitor system
4. Unauthorized Access

Issue: When the system detects an admin tries to view system activity logs to without being logged in or authorized.

Response: You must be logged in to reserve a parking slot. Please login to your account.

1. Network Failure

Issue: When admin has connectivity issues while monitoring system activity log.

Response: Network connection lost. Please check your internet connection and try again.

1. Suspicious Activity Detected

Issue: Unusual spikes in traffic or activity logs suggest potential misuse or automated attacks.

Response: Suspicious activity has been detected. The system is temporarily restricting access while investigating.

1. Effect of Parameters
2. Access Role Configuration

Problem: Admin roles are not properly defined, allowing unauthorized access to sensitive system functions or restricting necessary access.

Solution: Define role-based permissions clearly. Implement role management logic to restrict access based on responsibilities.

1. Real-Time Monitoring Interval

Problem: Monitoring data refreshes too slowly, causing delays in detecting and responding to issues.

1. Relationship of output to input
2. Input / Output Sequence
3. Admin access to system activity page.
4. System requests the data from the server.
5. Server sends GET requests to database.
6. If the system activity data exists, the server fetches the data to the system.
7. System displays all the system activity data to admin.

3.1.17 Admin Technical Support

**Sequence Diagram**



Figure 3.1.17 Sequence Diagram for Admin Technical Support

1. Authentication
2. Ensure the admin is logged into the system.
3. Response Abnormal Situation during technical support
4. System Crash During Troubleshooting

Issue: While resolving a user issue, the system crashes or throws an internal error.

Response: An unexpected error occurred during the troubleshooting process. The incident has been logged, and our technical team is investigating. We will update you as soon as possible.

1. Incomplete User Information

Issue: User submits a support request without providing sufficient details.

Solution: Your request lacks some necessary details. Please provide additional information such as error messages or steps taken, so we can assist you better.

1. Unauthorized Access

Issue: When the system detects an admin tries to review report issue details without being logged in or authorized.

Response: You must be logged in to reserve a parking slot. Please login to your account.

1. Network Failure

Issue: When admin has connectivity issues while accessing support tools or review report issue.

Response: Network connection lost. Please check your internet connection and try again.

1. Effect of Parameters
2. Support Admin Availability

Problem: Limited staff availability during peak times causes support backlog and delayed issue resolution.

Solution: Use shift planning and resource management tools to ensure support coverage during high-demand periods. Consider using chatbots or self-help resources for basic queries.

1. Support Tool Access Level

Problem: Admins do not have access to necessary diagnostic tools, delaying issue resolution.

Solution: Grant appropriate access levels to support tools based on roles. Ensure senior admins can access logs, user histories, and system diagnostics securely.

1. Relationship of output to input
2. Input / Output Sequence
3. User report issue to system.
4. System notify admin the report issue.
5. Admin request the system to review the report issue.
6. System displays the details of report issue to admin.
7. Admin solves the issue in the system.
8. If the issue is solved, Admin will send the support guide to the user.
9. If the user receives the support and the issue is solved, the user will send the confirm resolved to admin.
10. Admin updates the system logs.

3.1.18 Admin Implement and Maintain Security Measure

**Sequence Diagram**



Figure 3.1.18 Sequence diagram for Admin Implement and Maintain Security Measure

1. Authentication
2. Ensure the admin is logged into the system.
3. Response Abnormal Situation during implement and maintain security measure
4. Encryption Process Fails

Issue: System encryption fails due to a corrupted encryption module or configuration error.

Response: Encryption failed due to a system error. The incident has been logged, and the administrator has been notified. Please try again later.

1. Access Control Misconfiguration

Issue: Admin misconfigures access rules, accidentally locking out valid users.

Response: Access control update could not be applied due to conflicting rules. Please review and correct the configuration.

1. Logging System Unavailable

Issue: The system is unable to store logs during the security update process.

Response: Security actions completed, but logging failed. Retry log export or check system logger connection.

1. Effect of Parameters
2. Access Level Configuration

Problem: Security features like patching and access control are available to all users.

Solution: Restrict security operations to verified admin accounts only. Assign roles to enforce least privilege and prevent unauthorized access to sensitive controls.

1. Encryption Algorithm Type

Problem: Use of outdated or weak encryption algorithms puts user data at risk.

Solution: Implement strong, industry-standard encryption. Regularly review and upgrade encryption standards to stay ahead of emerging threats.

1. Patch Update Frequency

Problem: Infrequent or manual security updates expose the system to known vulnerabilities.

Solution: Define and automate regular patch cycles. Schedule monthly updates and urgent patch deployments for critical vulnerabilities.

1. Audit Logging Level

Problem: Lack of detailed logging hinders investigation during a security incident.

Solution: Enable verbose logging for all security operations. Store logs securely and ensure tamper-proof access for auditing.

1. Relationship of output to input
2. Input / Output Sequence
3. Admin needs to login to the security system.
4. System verifies the admin access.
5. Admin initiates a security update, triggering the system of the latest patches.
6. System applies patches and confirms successful update or logs errors if encountered.
7. Admin configures access control policies, such as updating roles, permissions, and user access rights
8. Access Control module applies the updates and confirms that configurations are in effect.
9. Admin enables encryption settings and ensures sensitive data is protected using secure algorithms.
10. System encrypts necessary data and confirms encryption status.
11. All actions are logged in the system audit trail, including timestamp, changes made, and actor identity.
12. The system generates a summary report confirming successful security configuration and log entries.
13. Admin reviews and acknowledges the confirmation.

## 3.2 Performance Requirements

Table 3.2.1 Table for Campus Ride-Sharing Platform Available Parking Time

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_P1 | The system shall provide real-time parking availability updates with a refresh interval of no more than 10 seconds. | Medium | Wee Jia Sheen |

Table 3.2.2 Table for Campus Ride-Sharing Platform Reservation Response Time

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_P2 | The system shall process parking or ride reservations within a maximum of 2 seconds after the user submits a request. Backend optimization and caching strategies will be implemented to meet this requirement. | Medium | Wee Jia Sheen |

Table 3.2.3 Table for Campus Ride-Sharing Platform Request/Offer Response Time

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_P3 | The system shall respond with suitable matches within 3 seconds. | Medium | Wee Jia Sheen |

Table 3.2.4 Table for Campus Ride-Sharing Platform Digital ID Verification

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_P4 | The digital ID verification process, used to validate whether a user is a current student, faculty, or staff member, must be completed within 2 seconds from the time of request. | Medium | Wee Jia Sheen |

## 3.3 Usability Requirements

The Campus Ride-sharing Platform shall complete the tasks given by users successfully with a high completion rate. This is important for ensuring interactions between users and systems are seamless and error-free. By consistently monitoring the system, we shall ensure that users can accomplish their tasks correctly within the platform with fast response time for a better experience.

The platform shall also be fulfilling the users’ satisfaction. To achieve this, there will be several measurement criteria that need to be fulfilled. The platform shall maintain a consistent layout and visual style across all pages to reduce load and increase familiarity. Additionally, the platform shall provide an intuitive user interface that allows users to easily navigate each feature. User satisfaction can be measured by user retention, the percentage of users returning to use the system regularly. By monitoring these metrics, the system is able to meet users’ needs and enhance satisfaction, maintaining positive user experience.

To prevent any harm to users when using the platform, there will be several measures taken to achieve this objective. First and foremost, we’re maintaining strict data privacy and security measures to protect users’ data from breaches. Secondly, making accessibility a priority ensures that all users, including individuals with disabilities, can effectively use the system, thereby fostering inclusivity and equal access to its features. Through ongoing monitoring and attention to these factors, the system can avoid harmful outcomes and create a safe, inclusive, and ethically responsible environment for alumni engagement and interaction.

## 3.4 Logical Database Requirements

3.4.1 User Table

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Description | Data Type | Key |
| User\_ID | Unique identifier for each user in the system. It differentiates students, staff, and admin | Int | PK |
| Name | Name of user | Char(100) |  |
| Email | Email address used for login | String |  |
| Password | Encrypted password used for user authentication | String |  |
| Role | Role of the user (e.g., student, staff,admin) | String |  |

3.4.2 Carpool Table

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Description | Data Type | Key |
| Carpool\_ID | Unique identifier for each carpool ride | Int | PK |
| Driver\_ID | User ID of the person offering the carpool | Int | FK |
| Destination | Destination of the trip | String |  |
| Departure\_Time | Scheduled time of departure | DateTime |  |
| Available\_Seats | Number of seats still available | Int |  |

3.4.3 Carpool Request Table

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Description | Data Type | Key |
| Request\_ID | Unique ID for each carpool request | Int | PK |
| Carpool\_ID | Carpool ride being requested | Int | FK |
| Requester\_ID | ID of user making the request | Int | FK |
| Request\_Time | Timestamp when request made | DateTime |  |
| Status | Status of request (pending, accepted, rejected) | String |  |

3.4.4 Parking Slot Table

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Description | Data Type | Key |
| Slot\_ID | Unique ID of the parking slot | Int | PK |
| Faculty | Description of the parking slot | String |  |
| Reserved\_By | User ID who reserved the slot, null if available | Int | FK |
| Status | Status of the slot (available/reserved) | String |  |

3.4.5 Booking Table

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Description | Data Type | Key |
| Booking\_ID | Unique ID for each booking | Int | PK |
| User\_ID | ID of the user who made the booking | Int | FK |
| Reference\_ID | Refers to Carpool\_ID or Slot\_ID | Int |  |
| Booking\_Time | Date and time of the booking | DateTime |  |

3.4.6 Admin Activity Log Table

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Description | Data Type | Key |
| Log\_ID | Unique identifier for the activity log | Int | PK |
| Admin\_ID | ID of the admin who performed the action | Int | FK |
| Action | Description of the activity performed (eg update slot) | String |  |
| Target\_ID | ID of the affected user (if applicable) | Int |  |
| Timestamp | Date and time of the action | DateTime |  |

3.4.7 Vehicle Table

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Description | Data Type | Key |
| vehicle\_id | Unique ID for the vehicle | Int | PK |
| user\_id | Owner | Int | FK |
| type | Vehicle type | Varchar |  |
| plate\_number | Vehicle plate number | Varchar |  |
| model | Vehicle model | Varchar |  |
| color | Vehicle color | Varchar |  |
| capacity | Number of available seats | Int |  |

3.4.8 Notification Table

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes | Description | Data Type | Key |
| notification\_id | Unique notification ID | int | PK |
| user\_id | Recipient user ID | Int | FK |
| title | Brief title of the notification | Varchar |  |
| message | Detailed message content | Varchar |  |
| type | Category (e.g., hardware, profile, parking) | Varchar |  |
| status | unread / read / archived | Varchar |  |
| created\_at | When it was created | Datetime |  |



3.4.8 ERD Diagram

## 3.5 Interface Requirements

3.5.1 User Authentication Interface

The system should provide secure login and registration interface for users using their campus email address. To clarify with this, the login screen is required to include with input fields (email & password), as well as the “Forget Password” feature for password recovery. The interface ensures only campus student and staff can access the platform to enhance security and trust within the campus community.

3.5.2 Ride Request & Offer Interface

The main function of the application is to allow users request or offer rides. The interface for request or offer should have form fields where it allow them to fill up either choose pick-up location and drop-off location, departure time. It should also integrate with real-time location access to visualize the route and display map within the application. With this feature, it helps users quickly understand available ride options and facilities effective ride-matching.

3.5.3 Mobile-Friendly and Responsive Design

Both student and staff will access the application via smartphone, similar to how they used “Uber” to call for a ride. Therefore, the interface must be optimized for mobile style, meaning that the interface such as button, text, size, and more should scale according to the mobile type (screen size and brand) and the performance also needs to optimize ensuring it has fast-loading. A responsive design capable of improving the accessibility and provide smooth experience for the user.

## 3.6 Design Constraints

3.6.1 Data Privacy Regulations

Compliance with data privacy laws, especially Malaysia’s Personal Data Protection Act (PDPA), is a primary design constraint in the development of this ride-sharing platform. To uphold students and staff members privacy rights, the system must obtain informed and explicit consent from users prior to collecting any personal information such as names, student IDs, vehicle information, or location data. This consent must be clearly communicated, and users should be able to control how their data is used.

Data access controls must be implemented to ensure that sensitive data is only accessible to authorized personnel. This involves adopting role-based access control (RBAC), encryption of sensitive information, and maintaining access logs to monitor usage and detect suspicious activities. For instance, only admin users should be able to view system- wide analytics or modify platform settings, while students and staff are restricted to their own profiles and ride interactions.

3.6.2 Integration with Existing Systems

The ride-platform must integrate with university-managed systems, including the authentication system (SSO), and real-time parking API. These integrations are dependent on stable and well-documented APIs. Any change in these systems may affect platform performance and must be handled well.

Synchronization between systems (user verification and digital ID status) must be real-time to ensure accurate permission management. As these services are not under the direct control of the platform developers.

3.6.3 Accessibility Standards

The platform must conform to the Web Content Accessibility Guidelines (WCAG) 2.1 to ensure usability for individuals with disabilities and to promote inclusiveness across the university community. This includes providing alternative text for icons, offering text labels for input fields and ensuring the interface is readable by screen readers for users. Buttons, forms, and menus should be properly labeled and navigable using standard tab orders.

The systems must also support color contrast and offer flexible font sizing and spacing to support users with low vision and poor eyesight. These practices ensure the platform is not only accessible but also comfortable to use for a broad range of users.

3.6.4 Mobile Device Constraints

The platform must be designed to function smoothly on a wide range of mobile devices, including both Android and iOS smartphones. Mobile devices have smaller screens, so the user interface must be kept simple, minimal text, and clear icons. Network connectivity on mobile is less reliable than on desktop environments. The system must be designed to handle temporary loss of internet connection gracefully, such as by caching data locally or displaying meaningful error messages when features are unavailable.

Mobile platforms have strict app store requirements. The design must comply with each platform’s guidelines for app behavior, security, privacy, and user experience to ensure approval and availability in official app stores. By addressing these mobile-specific constraints, the platform can deliver a better experience to students and staff who rely on their smartphones for daily transportation needs.

## 3.7 Software System Attributes

Several crucial software system features are necessary for the Campus Ride-Sharing Platform with Parking System Integration to guarantee that the system is not only fully functional but also maintainable.

Transportation systems that users rely on every day must be reliable. The platform must operate continuously without crashes or data problems because it handles actual parking data and carpool matching in real-time. Users may experience a considerable impact on their vacation plans as a result of downtime or mistakes. The system features real-time monitoring, and automated error recovery to ensure high reliability. System integrity will be maintained through data validation procedures. The platform is always available when needed, especially during busy campus hours.

Since the platform handles sensitive user data such as personal details, and parking reservations, security is a primary concern. To ensure the platform's security, the system will adhere to best practices in secure software design, such as role-based access control, secure SSO logins, and encrypted communications over HTTPS (TLS 1. 3). The platform will also fully comply with Malaysia's PDPA (Personal Data Protection Act). User data will be kept safe using AES-256 encryption. Additional security measures include routine penetration testing and the deployment of MFA for administrators.

The effectiveness of system interactions and the user experience are both affected by performance. Under typical loads, parking availability checks, ride confirmation, and carpool searches should be completed within 2 seconds to meet current performance benchmarks. To accomplish this, performance tuning will involve front-end caching, asynchronous back-end requests, and little server-side processing for each transaction. Stress testing and performance benchmarks will make sure that the system continues to respond even when it is used extensively.

User adoption depends on usability. The system should have a simple, user-friendly interface that students and employees can use with little assistance. It should take no more than five steps to complete tasks like booking parking or providing a ride. The user interface will adhere to university branding and design standards, ensuring visual consistency and trust. Actual users will be used in usability testing to collect feedback, pinpoint friction areas, and inform design enhancements. Additionally, accessibility features like readable fonts, keyboard navigation, and screen reader compatibility will be included.

Finally, interoperability is the system's capacity to communicate with outside platforms such as the university's student information system (SIS), parking management API, and digital ID verification system. The system uses OpenAPI 3. 0 standard for documentation and security. This enables campus systems and third-party services to interact with the platform in a way that is both efficient and secure.

The platform prioritizes these qualities—reliability, availability, security, maintainability, scalability, performance, portability, usability, and interoperability—to provide a complete, trustworthy, and innovative approach to the issues of campus mobility and parking.

## 3.8 Supporting Information



### Sample Input

Table 3.8.1 Student Sample Input

|  |  |
| --- | --- |
| **Student Profile** | |
| Name | Wee Jia Sheen |
| Student ID | 1211110222 |
| Phone No. | 013 – 200 4324 |
| Email | 1211110222@student.mmu.edu.my |
| Nationality | Malaysian |
| Address | Block J-02-03, Mutiara Ville, 63000 Cyberjaya, Selangor. |
| **Student Current Education** | |
| Specialization | Software Engineering |
| Year | Year 2 Sem 3 |
| **Student Offer Carpool** | |
| Student ID | 1211110222 |
| Message | Good morning! 😊I have class later at 12 PM — is anyone interested in carpooling with me? Let me know if you’d like to join!  Destination: Meet me at Mutiara Ville Lobby  Time: Will be there at 11.30AM |
| **Student View Available Parking** | |
| View | A map will be displayed with parking slots |
| Choice | Choose for FCI Faculty Parking Slots and will display available slots (green/red) |

Table 3.8.2 Student Sample Input

|  |  |
| --- | --- |
| **Student Profile** | |
| Name | Chia Kai Jun |
| Student ID | 1211111053 |
| Phone No. | 017 – 600 4783 |
| Email | 1211111053@student.mmu.edu.my |
| Nationality | Malaysian |
| Address | Block F-23-06, Mutiara Ville, 63000 Cyberjaya, Selangor. |
| **Student Current Education** | |
| Specialization | Data Science |
| Year | Year 1 Sem 1 |
| **Student Request Carpool** | |
| Student ID | 1211111053 |
| Message | I was wondering if I could join your carpool for today. Let me know if there’s space, thanks! |

Table 3.8.3 Staff Sample Input

|  |  |
| --- | --- |
| **Staff Profile** | |
| Name | Yap Wei Jian |
| Staff ID | MU243246 |
| Phone No. | 011 – 340 2526 |
| Email | MU243246@mmu.edu.my |
| Faculty | FCM |
| **Staff Reserve Parking** | |
| View | A map will be displayed with parking slots |
| Choice | Choose for FCM Faculty Parking Slots and choose the parking that want to reserve |
| **Staff View Available Parking** | |
| View | A map will be displayed with parking slots |
| Choice | Choose for FCM Faculty Parking Slots and will display available slots (green/red) |

### Validation Session

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Session ID** | **Date & Time** | **Technique** | **Section Reviewed** | **Participant & Role** | **No. of Defects** |
| VS\_01 | 15/6/2025; 8-10pm | Inspection | 1.0 Introduction - 5.0 Appendices | Whole Team | 16 |
| VS\_02 | 16/6/2025; 8-10pm | Walkthrough | 1.3 Product Overview | Whole Team | 5 |
| VS\_03 | 17/6/2025; 8-9pm | Walkthrough | 1.3 Product Overview, 3.1 Function | Whole Team | 6 |
| VS\_04 | 18/6/2025; 8-10pm | Inspection | 1.3 Product Overview, 3.4 Logical Database Requirements | Whole Team | 4 |
| VS\_05 | 19/6/2025; 9-11pm | Inspection | 1.3 Product Overview, 3.1 Function, 3.2 Performance Requirements | Whole Team | 5 |
| VS\_06 | 20/6/2025;10-11pm | Inspection | 3.1 Function, 3.2 Performance Requirements | Whole Team | 5 |

### Defect Summary

1. **Content Summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req ID** | **Validation & Defect Description** | **Detected By** | **Comment / Suggested Fix** | **Session ID** | **Severity** |
| REQ-GUI1 | Color codes not accessible | Xuan Lin | Add WCAG-compliant contrast options | VS\_05 | 2 |
| REQ\_GUI2 | “User‑friendly layout” is subjective | Ka Sheng | Define measurable criteria (e.g. ≤ 2 clicks for key actions, ≤ 10 % drop‑off) | VS\_02 | 3 |
| REQ\_GUI3 | “Responsive design” lacks breakpoint definitions | Ka Sheng | Specify device width ranges (e.g. mobile ≤ 480 px, tablet 481–1024 px) | VS\_02 | 3 |
| REQ\_H2 | “Latest and three previous major versions” vague | Ka Sheng | List explicit OS versions (iOS 17–14, Android 14–11) | VS\_02 | 3 |
| REQ\_S1 | Specifies SQLite despite scalability needs | Kelven | Replace with scalable RDBMS (PostgreSQL) or justify SQLite prototype | VS\_01 | 4 |
| REQ\_C3 | Duplicate communication requirement (REQ\_C1 & REQ\_C3 identical) | Kelven | Remove or merge duplicate; renumber following entries | VS\_01 | 2 |
| REQ\_R1 | No MFA requirement for admin login | Kelven | Add 2FA for admin login | VS\_01 | 5 |
| REQ\_P2 | Reservation delay > 2s | Xuan Lin | Optimize backend response | VS\_06 | 3 |
| 1.3.2 | “Register Vehicle” missing in use‑case diagram | Xuan Lin | Modify use‑case diagram | VS\_04 | 5 |
| 1.3.4.2 | Scalability not mentioned for peak traffic | Shazreen | Add: “Support 500 + users concurrently” | VS\_04 | 4 |
| 3.1 | Notification system not defined | Ka Sheng | Define booking and admin notifications | VS\_04 | 5 |
| 3.4.7 | Vehicle table lacks vehicle type | Kelven | Add “vehicle\_type” column to DB schema | VS\_04 | 3 |

1. **Documentation Defect**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Page No.** | **Validation & Defect Description** | **Detected By** | **Comment / Suggested Fix** | **Session ID** | **Severity** |
| 1-89 | Manual spacing used in place of page breaks, causing layout instability | Kelven | Replace multiple consecutive paragraph breaks with appropriate page or section breaks | VS\_01 | 5 |
| 1-89 | Inconsistent paragraph formatting: Each paragraph is terminated with a manual line break (Enter key), even when the content should flow continuously | Kelven | Remove unnecessary paragraph breaks and restructure content into coherent paragraphs | VS\_01 | 5 |
| 2&3 | Outdated Table of Content | Shazreen | Update the Table of Content with latest information | VS\_01 | 3 |
| 10 | Inconsistent content spacing | Ka Sheng | Standardize paragraph and heading spacing | VS\_01 | 1 |
| 13 | Use of “staffs” instead of “staff” | Shazreen | Correct to “staff” | VS\_03 | 2 |
| 16 | Heading spacing inconsistent (Table 1.3.3 User Characteristics) | Ka Sheng | Fix heading indentation | VS\_01 | 1 |
| 16 | Missing user accessibility notes | Xuan Lin | Add WCAG 2.1 compliance | VS\_05 | 2 |
| 31 | Typo in Route and Time Selection at Effect of Parameter: "mathing" | Kelven | Correct to "matching" | VS\_03 | 2 |
| 39 | Inconsistent capitalization | Shazreen | Capitalize the words | VS\_03 | 2 |
| 44 | Figure 3.1.8 numbered twice | Ka Sheng | Rename duplicate figure to maintain unique numbering | VS\_05 | 3 |
| 55 | Typo in Profile Completeness at Effect of Parameter: "Imcomplete" | Kelven | Correct to "Incomplete" | VS\_03 | 2 |
| 56 | Sequence diagram without label at 3.1.13 | Shazreen | Add label "Figure 3.1.13 Sequence Diagram for Admin Manage Users" | VS\_03 | 2 |
| 57 | Inconsistent punctuation in Response Abnormal Situation during monitor sytem | Shazreen | Fix the punctuations | VS\_03 | 2 |
| 66 | Outdated performance metric | Xuan Lin | Update to current benchmark | VS\_06 | 2 |
| 71 | Typo in Carpool table header: “Deaparture\_Time” | Xuan Lin | Correct to “Departure\_Time” | VS\_01 | 2 |
| 75 | Format inconsistency: heading 3.6.2 spacing differs from 3.6.1 & 3.6.3 | Ka Sheng | Standardize heading formatting across sub‑sections | VS\_01 | 1 |
| 78 | Inconsistent terminology: “university student information system (SIS)” vs “university database” | Ka Sheng | Standardize to “university student information system (SIS)” throughout | VS\_05 | 2 |

1. **Agreement Defect**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req ID** | **Validation Description / Stakeholder Concern** | **Mismatch Description** | **Detected By** | **Session ID** | **Severity** |
| REQ\_S3 | Social login not aligned with campus security policy | OAuth vs SSO‑only requirement | Kelven | VS\_01 | 5 |
| REQ\_P1 | 5s refresh rate not feasible | Technical feasibility gap | Xuan Lin | VS\_06 | 4 |
| REQ\_P4 | Staff can change parking anytime vs admin preference for tighter control | Over‑flexibility vs control | Ka Sheng | VS\_01 | 4 |

### Conflict Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Conflict ID** | **Conflict Description** | **Conflict Analysis** | **Stakeholders Involved** | **Session ID** |
| CF\_01 | OAuth login vs campus SSO‑only policy | Campus policy prohibits OAuth; must use secure SSO | Dev Team, IT Admin | VS\_01 |
| CF\_02 | Staff can change parking anytime vs admin restrictions | Risk of misuse; requires guardrails | Staff, Admin | VS\_01 |
| CF\_03 | UI lacks mobile mockups | Developers unclear on mobile layout expectations | Dev, QA | VS\_02 |
| CF\_04 | Missing insurance validation | Liability concern if unverified drivers | Legal, Dev | VS\_03 |
| CF\_05 | Account deletion not possible | Violates PDPA/data protection expectations | Legal, Users | VS\_04 |
| CF\_06 | GPS tracking assumed, but no opt‑out | May violate privacy; need consent | Users, Admin | VS\_04 |
| CF\_07 | Ride history not logged | QA/stakeholders expect traceability/auditing | Admin, QA | VS\_04 |
| CF\_08 | Scalability unclear | Infrastructure may fail under real load | Infra, Dev | VS\_04 |
| CF\_09 | Notification system missing | User expectations not met | Users, UX | VS\_04 |
| CF\_10 | Vehicle info lacks vehicle type | Matching/filtering failures | Dev, QA | VS\_04 |
| CF\_11 | No abuse reporting system | Trust/safety risk without moderation | Users, Admin | VS\_05 |
| CF\_12 | Staff parking without availability check | Overbooking risk | Infra, Admin | VS\_05 |
| CF\_13 | Parking reservation limits for staff unclear | Staff want guaranteed slots; system lacks capacity | Staff, Admin | VS\_05 |
| CF\_14 | Performance vs Accessibility | Developers want speed; UX team wants full WCAG compliance | Dev Team, UX, QA | VS\_05 |
| CF\_15 | Real-time vs system load | Real-time updates strain backend during peak hours | Dev Team, QA, PO | VS\_06 |

### Conflict Analysis Resolution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Conflict ID** | **Resolution Strategy** | **Resolved (Y/N)** | **Outcome** | **Justification** |
| CF\_01 | Replaced OAuth with campus SSO login | Y | Compliant with campus policy | Aligns with IT standards |
| CF\_02 | Added one‑change‑per‑day rule with admin override | Y | Balanced user flexibility & oversight | Prevents abuse without over‑restriction |
| CF\_03 | Created UI wireframes for both desktop dashboard & mobile interface | Y | Clarified design expectations | Aids frontend development |
| CF\_04 | Required insurance upload for drivers | Y | Legally compliant | Ensures driver validity |
| CF\_05 | Added user account deletion feature | Y | PDPA/GDPR‑compliant | Respects data subject rights |
| CF\_06 | Added opt‑out toggle for GPS tracking | Y | Consent‑based privacy model | Improves user trust |
| CF\_07 | Added ride participation logging table | Y | Enables auditing | Supports traceability |
| CF\_08 | Specified scalability requirement (support 500 + concurrent users) | Y | Infrastructure readiness confirmed | Mitigates load‑related failures |
| CF\_09 | Included notification system in SRS | Y | Enhanced UX | Meets real‑world usage patterns |
| CF\_10 | Added vehicle\_type field in registration & DB schema | Y | Supports ride matching/filtering | Enables logical matching |
| CF\_11 | Added abuse report feature routed to admin | Y | Enhanced platform safety | Enables moderation |
| CF\_12 | Implemented parking availability check & reservation control | Y | Prevents overbooking | Matches physical resource limits |
| CF\_13 | Implemented daily reservation quotas for staff | Y | Staff guaranteed 2 slot/day | Ensures equitable distribution |
| CF\_14 | Prioritized accessibility for core features | Y | WCAG compliance added to login and dashboard | Balanced usability with performance trade-offs |
| CF\_15 | Backend optimization + caching | Y | Reduced refresh rate to 10s | Balanced performance and feasibility |

### Change Log

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Change ID** | **Req ID** | **Summary of Change** | **Proposed By** | **Date** | **Session ID** |
| CH\_01 | REQ\_R1 | Added 2FA for admin login | Kelven | 15/6/2025 | VS\_01 |
| CH\_02 | REQ\_S1 | Replace with scalable RDBMS (PostgreSQL) or justify SQLite prototype | Kelven | 15/6/2025 | VS\_01 |
| CH\_03 | REQ\_C3 | Removed or merged duplicate communication requirement | Shazreen | 15/6/2025 | VS\_01 |
| CH\_04 | REQ\_S3 | Replaced OAuth with campus SSO login | Kelven | 15/6/2025 | VS\_01 |
| CH\_05 | REQ\_H2 | List explicit OS versions (iOS 17–14, Android 14–11) | Ka Sheng | 16/6/2025 | VS\_02 |
| CH\_06 | REQ\_GUI3 | Specified device width ranges for responsive design | Ka Sheng | 16/6/2025 | VS\_02 |
| CH\_07 | REQ-GUI1 | Added WCAG contrast compliance | Xuan Lin | 19/6/2025 | VS\_05 |
| CH\_08 | REQ\_P1 | Reduced refresh rate to 10s for feasibility | Xuan Lin | 20/6/2025 | VS\_06 |
| CH\_09 | REQ\_P2 | Backend optimization for faster reservation | Xuan Lin | 20/6/2025 | VS\_06 |

### Requirement Traceability Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req ID** | **Requirement Description** | **Linked Goal(s)** | **Feature(s)** | **Use Case(s)** | **Traceability Score (1-4)** |
| REQ\_GUI2 | Dashboard layout | G2 | F1 | UC-02 | 3 |
| REQ\_P2 | Reservation <2s | G2 | F2 | UC-03 | 3 |

### Role in Requirements Validation, Negotiation & Management

|  |  |  |
| --- | --- | --- |
| **Student Name** | **Primary Responsibility** | **No. of Session Participated** |
| Kelven Yee Kai Wen | Content validation, traceability matrix updates | 2 |
| Koh Xuan Lin | GitHub version control, changelog maintenance, conflict logging | 2 |
| Ow Ka Sheng | Conflict analysis, stakeholder concern tracking | 2 |
| Shazreen Binti Sheridan | Documentation review, defect classification | 1 |

### Version Control & Configuration Summary

Repository Branch: project-part-2

Key files:

* SRS.md
* TT2L\_GF\_SRS.docx
* changelog.md

Commits made by Kelven:

Commits made by Koh Xuan Lin:

Commits made by Ow Ka Sheng:

Commits made by Shazreen:

Additional Metrics

Pull Requests made by Kelven:

Pull Requests made by Koh Xuan Lin:

Pull Requests made by Ow Ka Sheng:

Pull Requests made by Shazreen:

Change Log Entries made by Kelven:

Change Log Entries made by Koh Xuan Lin:

Change Log Entries made by Ow Ka Sheng:

Change Log Entries made by Shazreen:

# 4.0 Verification

## 4.1 Specified Requirements

Table 4.1 Specified Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requirement ID | Requirement | Verification Method | Responsibility | Event-based Timing | Venue/ Environment |
| REQ\_R1 | Users log in with campus credentials | Functional Testing | QA Team | During login attempt | Test Environment |
| REQ\_R2 | Verified students/staff can access booking features | Integration Testing | QA Team | After login verification | Test Environment |
| REQ\_R3 | Users post and view carpool | Use Case Testing | QA Team | Listing | Test Environment |
| REQ\_R4 | Staff reserve campus parking | Functional Testing | Development Team | Parking Reservation | Development Environment |

## 4.2 External Interfaces

Table 4.2 External Interfaces

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Interface ID | Description | Verification Method | Responsibility | Event-based Timing | Venue/ Environment |
| INT\_E1 | University (SSO) for login | Integration Testing | Development Team | During login page | Development Environment |
| INT\_E2 | Real-time Parking Availability API | API Response Testing | Development Team | Parking page | Development Environment |
| INT\_E3 | Digital Student ID Verification System | Credential Sync Testing | QA Team, Development Team | During login page | Identify Management API |

## 4.3 Functions

Table 4.3 Functions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Function ID | Function | Verification Method | Responsibility | Event-based Timing | Venue/ Environment |
| FUNC\_F1 | Offer Ride | Use Case Walkthrough | QA Team | During offer submission | Offer Ride Page |
| FUNC\_F2 | Request Ride | Functional Testing | QA Team | After joining request | Listing View Page |
| FUNC\_F3 | Reserve Parking | Validation Testing | Development Team | During request | Staff Parking Portal |
| FUNC\_F4 | View and accept join request | Workflow Testing | QA Team | Carpool detail view | Test Environment |

## 4.4 Usability Requirements

Table 4.4 Usability Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requirement ID | Requirement | Verification Method | Responsibility | Event-based Timing | Venue/ Environment |
| REQ\_U1 | User-Friendly Interface | Usability Testing with user portal | UX Team | During UI/UX design phase | Usability Lab |
| REQ\_U2 | Accessible Design | Accessibility Testing using Tools like aXe | QA Team | Post-UI development | Test Environment |
| REQ\_U3 | Efficient Navigation | Task Analysis and User Testing | UX Team | After initial UI setup | Usability Lab |
| REQ\_U4 | Error Prevention & Handling | Review and Test Error | QA Team | During functional testing | Test Environment |

## 4.5 Performance Requirements

Table 4.5 Performance Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requirement ID | Requirement | Verification Method | Responsibility | Event-based Timing | Venue/ Environment |
| REQ\_P1 | Response Time | Load testing and Speed Benchmarking | QA Team | After development | Performance Testing Lab |
| REQ\_P2 | Updates Refresh | API Monitoring | Development Team | Live dashboard | Parking Module |
| REQ\_P3 | Data Processing Speed | Testing data handling and processing times. | QA Team | During performance testing | Performance Testing Lab |

## 4.6 Logical Database Requirements

Table 4.6 Logical Database Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Description | Data Type | Key |
| user\_id | ID for each registered user | Int | PK |
| carpool\_id | Carpool offering ID | Int | PK |
| request\_id | Request made by a user | Int | FK |
| parkingSlot\_id | ID of reserved parking slot | Int | FK |
| user\_role | Role of user (student, staff) | String |  |
| timestamp | Date and time of action | DateTime |  |

## 4.7 Design Constraints

Table 4.7 Design Constraints

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Constraint ID | Constraint | Verification Method | Responsibility | Event-based Timing | Venue/ Environment |
| CON\_D1 | Use of Standard APIs | Review API documentation and compliance | Development Team | During integration phase | Development Environment |
| CON\_D2 | Mobile Compatibility | Cross-platform testing on mobile devices | QA Team | Post-development | Mobile Testing Lab |

Table 4.7 specifies constraints on the system design imposed by external standards, regulatory requirements, or project limitations. These constraints must be considered during the design and development phases to ensure compliance and feasibility.

## 4.8 Standards Compliance

Table 4.8 Standards Compliance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Standard ID | Standard | Verification Method | Responsibility | Event-based Timing | Venue/ Environment |
| STD\_S1 | PDPA | Compliance Testing | Security Team | Post-development review | Test Environment |
| STD\_S2 | ISO 27001 | Quality assurance based on ISO standards | QA Team | Throughout development | QA Lab |
| STD\_S3 | WCAG 2.1 | Accessibility Tool Test | QA Team | During UI design |  |

## 4.9 Software System Attributes

Table 4.9 Software System Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute ID | Attribute | Verification Method | Responsibility | Event-based Timing | Venue/ Environment |
| ATT\_A1 | Reliability | Continuous monitoring and error rate analysis. | QA Team | After deployment | Production Environment |
| ATT\_A2 | Availability | Uptime monitoring and failover testing. | Infrastructure Team | After deployment | Cloud Environment |
| ATT\_A3 | Maintainability | Code reviews and maintainability assessments. | Development Team | During development cycles | Development Environment |
| ATT\_A4 | Portability | Cross-platform testing on different systems | QA Team | After development | Test Environment |

# 5.0 Appendices

## 5.1 Assumptions and Dependencies

Assumptions

1. The development team will have consistent access to the university’s parking system and digital ID verification service. These systems need to be up and running, with reliable APIs, we can integrate them smoothly.
2. Students and staff being open to the idea of ride-sharing and willing to use the platform as part of their regular commute.
3. The project needs university admin to actively back the launch, promotion, and ongoing use of the platform across campus.
4. Users are expected to know how to use a mobile or web app things like logging in, navigating screens, and filling out simple content.

Dependencies

1. A stable internet connection is needed across campus and in users’ areas, especially for real-time updates, route planning, and map features.
2. The system relies on the university’s login service (e.g., SSO) and parking data APIs. These need to stay reliable and well-documented for long-term use.
3. We need continued access to tools and frameworks that help us meet data protection standards like PDPA and ISO 27001.

## 5.2 Acronyms and Abbreviations

Admin - Administrator

API - Application Programming Interface

HTTPS - Hypertext Transfer Protocol Secure

CRM - Customer Relationship Management

OAuth - Open Authorization login protocol

PDPA - Personal Data Protection Act (Malaysia)

ISO - International Organization for Standardization

SSO - Single Sign-On

RBAC - Role-Based Access Control

WCAG - Web Content Accessibility Guidelines

QA - Quality Assurance

UI/UX - User Interface/ User Experience

Int - Integer

String - String Data Type

Char - Character

PK - Primary Key

FK - Foreign Key

## 5.3 Definition

GUI Graphical User Interface

RAM Random Access Memory

SSD Solid State Drive