### **Lab 1 Deliverables (Group 5)**

Members: Dino Edouard Halley Yacat (U2340094L), Aniston Tan (U2340956G),

Tyler Goh (U2221894H), Ryan Paulo Perez Mendoza (U2340781H), Samantha Ho (U2340103B), Ng Zi En Chloe (U2340186C)

### **Project Statement**

App Name: BikeParkWhere

Our team plans to **develop a system that shows the available parking spots to Cyclists**.

Secondary function: Community update feature to update number of vacancies available.

We aim to make finding bike parking spots easier for them.

Target Audience: Cyclists with personal bikes

API: Bicycle parking

### **Functional Requirements**

1. Users shall be able to login to the App.
   1. The App shall allow users to register and log in with secure authentication.
      1. Users must be able to reset their password.
      2. Users must be authenticated with the system’s user database.
      3. If the user is not found in the database, the system must not allow the user to login.
      4. If the user fails to login, the system shall prompt the user to re-enter its username and password.
2. Users shall be able to turn on their location services.
   1. Users can either enter their destination or use their current location.
   2. This will lead the user to a new page showing parking space availability. The page must display :
      1. The top 3 nearest Parking Spaces
      2. The Address of the Parking Spaces
      3. The distance and route from the user’s destination/current location to the Parking Space
   3. There shall be a prompt/feature to reserve parking space.
3. Users shall be able to reserve a parking spot at the parking space of their choice.
   1. Users shall be able to view the available spaces
      1. The system must fetch the real-time vacancy data of the top 3 nearest parking spaces to the user.
         1. The vacancy data displayed for the selected parking spaces must consist of the total parking spaces available.
         2. The vacancy data displayed for the selected parking space must consist of the type of parking space (sheltered/not sheltered).
   2. Users can only reserve a parking spot if the number of available parking spots is more than 0.
   3. If the top 3 nearest parking spots are all full, the system will fetch the next 3 nearest parking spots.
   4. Upon a successful reservation, the system must update the vacancy data of the parking space.
      1. The available parking spots at the parking space must decrease by one.
      2. The status of the reservation shall be indicated as “Reserved.” and the Parking Spot will not be available for “Reservation” by another user.
      3. Upon arrival, the status of the reservation shall be updated to “Arrived.”
   5. The system shall hold a reservation for at most 30 minutes.
      1. The system shall make the parking spot available if the user fails to arrive after 30 minutes.
   6. Each user must have at most one reservation at any time.
4. Users shall be able to cancel their reservation.
   1. The system must update the vacancy data of the Bicycle Parking Space upon a successful cancellation.

### Non-Functional Requirements

| Usability | Languages   * The app shall support multiple languages based on user’s selected language preference * Help messages must be displayed in the user’s preferred language |
| --- | --- |
| Accessibility   * The app should be accessible to users with disabilities (e.g screen readers) |
| Responsive User Interface   * The app shall have an intuitive interface with a response time of 5 seconds or less for common tasks |
| Reliability | Application Launch   * The app should have a 99.9% uptime * After a system reboot/launch, the full system functionality must be restored within 5 minutes |
| Performance | Data Retrieval Efficiency   * The app should efficiently handle real-time data updates with a latency of less than 5 seconds * Map rendering and data visualisation should load within 3 seconds |
| Supportability | The database must be replaceable with any commercial product supporting standard SQL queries |

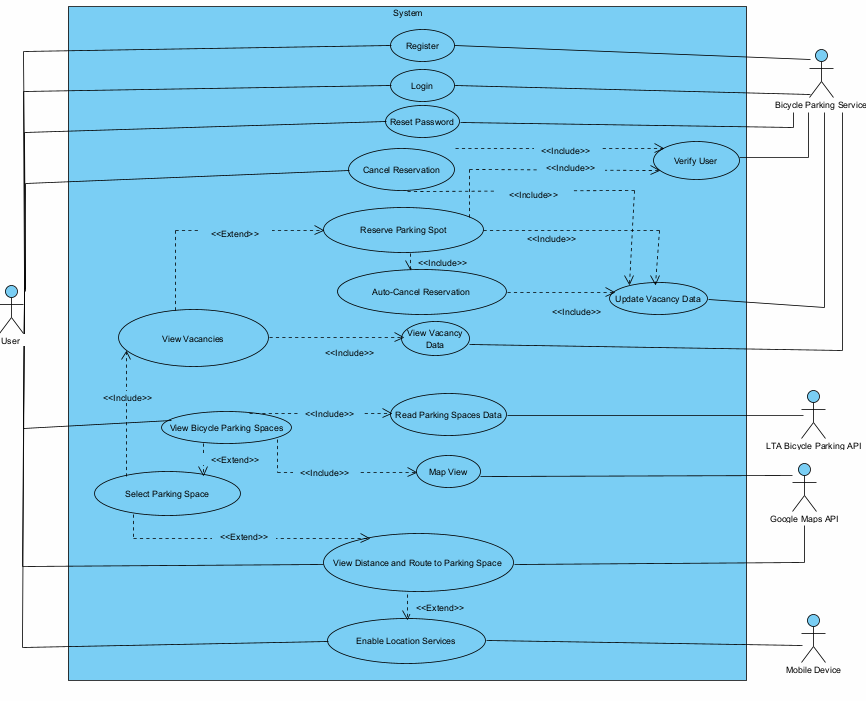
### 

### Data Dictionary

| Term | Definition |
| --- | --- |
| Destination | A location entered by a user as their intended place of arrival |
| Parking Space | A designated area where users can park their bicycles |
| Available Parking Spaces | The number of unoccupied parking spaces at a given parking location |
| Sheltered Parking Space | A parking space that is covered or located indoors |
| Unsheltered Parking Space | A parking space that is open and exposed to weather conditions |
| Route | The path a user must take to reach the parking space from their current location |
| Distance | The estimated measured length from the user’s current location to a parking space |

### 

### **Use Case Diagram**



### 

### **Use Case Description**

| Use Case ID: | UC1 | | |
| --- | --- | --- | --- |
| Use Case Name: | User Registration | | |
| Created By: | Ryan Paulo | Last Updated By: | Tyler |
| Date Created: | 08/02/2025 | Date Last Updated: | 10/02/2025 |

| Actor: | User  Bicycle Parking Service |
| --- | --- |
| Description: | Allows users to create a new account to access the system |
| Preconditions: | The user is not registered in the system |
| Postconditions: | The user account is created and stored in the user database  The user can now log in |
| Priority: | High |
| Frequency of Use: | Occasionally (when a new user registers) |
| Flow of Events: | 1. The system prompts the user to enter a username, password and registration details 2. The system validates the provided information 3. The system stores the user’s credentials in the database 4. The system confirms the successful registration |
| Alternative Flows: | If the user input is invalid (missing fields)   1. The system notifies the user 2. Return to step 1 |
| Exceptions: | If the username is already taken   1. The system notifies the user and asks for a new one.   If profile is already registered   1. The app prompts the user to log in instead. |
| Includes: | Nil |
| Special Requirements: | Password meets security criteria (minimum length of 10 characters, include numbers and special characters) |
| Assumptions: | Users provide valid registration details |
| Notes and Issues: | Ensure secure password storage |

| Use Case ID: | UC2 | | |
| --- | --- | --- | --- |
| Use Case Name: | User Login | | |
| Created By: | Ryan Paulo | Last Updated By: | Tyler |
| Date Created: | 08/02/2025 | Date Last Updated: | 10/02/2025 |

| Actor: | User  Bicycle Parking Service |
| --- | --- |
| Description: | Allows users to log into the system using valid credentials. |
| Preconditions: | The user must be registered |
| Postconditions: | The user is authenticated and granted access. |
| Priority: | High |
| Frequency of Use: | Frequently (when the user accesses the system for the first time/chooses to log out) |
| Flow of Events: | 1. The user enters their username and password. 2. The system verifies credentials against the user database. 3. If the credentials match, the user is granted access. |
| Alternative Flows: | If the credentials do not match   1. The app prompts the user to re-enter their details. 2. Return to step 1. |
| Exceptions: | If the user does not exist in the database (the username cannot be found)   1. The app denies access. 2. The app prompts the user to register.   If the user fails to login five times in a row   1. The account is locked. 2. The user has to reset the account by resetting their password. |
| Includes: | Nil |
| Special Requirements: | Secure authentication methods should be used |
| Assumptions: | Users remember their credentials |
| Notes and Issues: | Implement password hashing |

| Use Case ID: | UC3 | | |
| --- | --- | --- | --- |
| Use Case Name: | Reset Password | | |
| Created By: | Ryan Paulo | Last Updated By: | Tyler |
| Date Created: | 08/02/2025 | Date Last Updated: | 10/02/2025 |

| Actor: | User  Bicycle Parking Service |
| --- | --- |
| Description: | Allows users to reset their password |
| Preconditions: | The user has an existing account |
| Postconditions: | The user's password is successfully updated |
| Priority: | Medium |
| Frequency of Use: | Occasionally |
| Flow of Events: | 1. The user requests password reset. 2. The system sends a reset link to email. 3. The user clicks the link and enters a new password |
| Alternative Flows: | If the user enters an unregistered email   1. The system asks the user to re-enter their identification details 2. Go to step 2   If the confirmation email is not received   1. The user requests password reset again 2. Go to step 1 |
| Exceptions: | If the reset link expires before use   1. The password reset request is terminated 2. The system asks the user to resend verification email |
| Includes: | Nil |
| Special Requirements: | Nil |
| Assumptions: | User has access to the registered email |
| Notes and Issues: | Consider multi-factor authentication |

| Use Case ID: | UC4 | | |
| --- | --- | --- | --- |
| Use Case Name: | Enable Location Services | | |
| Created By: | Ryan Paulo | Last Updated By: | Tyler |
| Date Created: | 08/02/2025 | Date Last Updated: | 10/02/2025 |

| Actor: | User, Device |
| --- | --- |
| Description: | The user enables location services to find nearby parking spaces. |
| Preconditions: | The user's device must support location services. |
| Postconditions: | The system gains access to the user's location. |
| Priority: | High |
| Frequency of Use: | Occasionally |
| Flow of Events: | 1. The user enables location services 2. The system requests permission 3. The user grants permission 4. The system accesses user location |
| Alternative Flows: | If the device location services are disabled   1. The system notifies the user and prompts the user to turn it on 2. Go to step 1 |
| Exceptions: | If the user does not grant permission   1. The system prompts manual location input.   If the GPS signal is unavailable   1. The system automatically refreshes until the signal is available |
| Includes: | Nil |
| Special Requirements: | Device used has GPS capabilities |
| Assumptions: | The user wants to grant location permissions |
| Notes and Issues: | Implement secondary option for manual location entry |

| Use Case ID: | UC5 | | |
| --- | --- | --- | --- |
| Use Case Name: | View Bicycle Parking Spaces | | |
| Created By: | Ryan Paulo | Last Updated By: | Tyler |
| Date Created: | 08/02/2025 | Date Last Updated: | 10/02/2025 |

| Actor: | User  Google Maps API  Bicycle Parking Service |
| --- | --- |
| Description: | Displays parking spaces near the user's location |
| Preconditions: | User is logged in and has enabled location services |
| Postconditions: | List of nearest three parking spaces is displayed |
| Priority: | High |
| Frequency of Use: | Frequently |
| Flow of Events: | 1. The user requests to view parking spaces 2. The system fetches parking data from external APIs 3. The system maps the parking locations on google map’s view 4. The system suggests parking spaces within the radius 5. IF the user selects a parking space, then Google Maps API displays a route and distance |
| Alternative Flows: | If API fails to return data   1. The system notifies the user and tries to re-establish the connection 2. Return to step 2 |
| Exceptions: | If there are no parking spaces available   1. The system asks the user to move to a new location before requesting again. |
| Includes: | Read Parking Spaces Data |
| Special Requirements: | Real-time updates |
| Assumptions: | APIs provide accurate data |
| Notes and Issues: | Implement a refresh feature |

| Use Case ID: | UC6 | | |
| --- | --- | --- | --- |
| Use Case Name: | Select Bicycle Parking Space | | |
| Created By: | Ryan Paulo | Last Updated By: | Tyler |
| Date Created: | 08/02/2025 | Date Last Updated: | 10/02/2025 |

| Actor: | User  Google Maps API  Bicycle Parking Service |
| --- | --- |
| Description: | Displays parking information of the selected parking space and allows for reservation |
| Preconditions: | User has selected a parking space |
| Postconditions: | Vacancy information is displayed |
| Priority: | High |
| Frequency of Use: | Frequently |
| Flow of Events: | 1. The system displays the amount of vacancies for the parking space 2. IF the user selects view route, then he uses the extended View Distance and Route to Parking Space use case 3. IF the user selects reserve parking, then he uses the Reserve Parking Spot use case |
| Alternative Flows: | If API fails to return data   1. The system notifies the user and tries to re-establish the connection 2. Return to step 2 |
| Exceptions: |  |
| Includes: | View Vacancies |
| Special Requirements: | Real-time updates |
| Assumptions: | APIs provide accurate data |
| Notes and Issues: | Implement caching for efficiency |

| Use Case ID: | UC7 | | |
| --- | --- | --- | --- |
| Use Case Name: | Reserve Parking Spot | | |
| Created By: | Ryan Paulo | Last Updated By: | Tyler |
| Date Created: | 08/02/2025 | Date Last Updated: | 10/02/2025 |

| Actor: | User  Bicycle Parking Service |
| --- | --- |
| Description: | Allows users to reserve a parking spot |
| Preconditions: | The user is logged in and there is at least one available spot |
| Postconditions: | Parking spot is reserved and vacancy data is updated |
| Priority: | High |
| Frequency of Use: | Occasionally (as needed by the user) |
| Flow of Events: | 1. The user selects a parking space and requests reservation 2. The system checks availability 3. If available, reservation is confirmed 4. Vacancy data is updated 5. When the user reaches the parking slot, they scan a QR code found at the space to confirm that they have reached 6. IF the user doesn’t scan the QR code within 30 minutes of the reservation, the reservation is cancelled |
| Alternative Flows: | If there are no available spots   1. The system prompts the user to pick another parking space 2. Return to step 2 |
| Exceptions: | If the QR code has not been scanned within 30 minutes of booking   1. The system auto-cancels the reservation 2. The user has to make another reservation |
| Includes: | Update Vacancy Data |
| Special Requirements: | Reservation expires if the user does not arrive within 30 minutes. |
| Assumptions: | User arrives on time |
| Notes and Issues: | Implement notifications |

| Use Case ID: | UC8 | | |
| --- | --- | --- | --- |
| Use Case Name: | Cancel Reservation | | |
| Created By: | Ryan Paulo | Last Updated By: | Tyler |
| Date Created: | 08/02/2025 | Date Last Updated: | 10/02/2025 |

| Actor: | User  Bicycle Parking Service |
| --- | --- |
| Description: | Allows users to cancel an active reservations |
| Preconditions: | User has an active reservations |
| Postconditions: | Parking spot is freed |
| Priority: | Medium |
| Frequency of Use: | Occasionally |
| Flow of Events: | 1. The user selects the cancellation option 2. The system verifies the user is logged in 3. The system processes the cancellation 4. Vacancy data is updated |
| Alternative Flows: | If the cancellation request fails   1. The system notifies the user 2. Return to step 1 |
| Exceptions: |  |
| Includes: | Update Vacancy Data |
| Special Requirements: | Real-time updates |
| Assumptions: | User intends to cancel |
| Notes and Issues: | Allow re-reservation if needed |

### **UI Mockup**

