

### **Use Case 1: Sign-up**

**Description:** The user signs up for a new account

**Actors:** User, Application System

**Pre-conditions:** The User has a mobile number and does not have an existing account

**Post-conditions:** The User creates an account successfully

**The flow of events:**

1. The application will prompt the User for his/her mobile number.
2. The User will input his/her 8-digit mobile number.
3. The application validates the mobile number.
4. Upon successful validation, the application will prompt the User for a 'password' and 'password confirmation'.
5. The User will enter the exact password for the 'password' prompt and 'password confirmation' prompt.
6. Once 'password' input and 'password confirmation' inputs are verified, the application will store the User's mobile number and password in the system.
7. The User is prompted for a security answer to a security question for future login verification purposes.
8. A new account will be created for the User upon successful completion of the security question.

**Alternative flows:**

AF-S1: If any of the fields are blank.

1. The application will display "Error. There cannot be an empty field."
2. The application will return to step 1.

AF-S2: If a mobile number is registered, or invalid.

1. The application will display "Error. The number is registered to an existing account."
2. The application will return to step 1.

AF-S2: If a mobile number is invalid.

1. The application will display "Error. Invalid number."
2. The application will return to step 1.

AF-S5: If the passwords do not meet the minimum length requirement.

1. The application will display "Error. Passwords must be at least 6 characters long."
2. The application will return to step 4.

AF-S5: If the passwords do not match.

1. The application will display "Passwords do not match. Please try again!"
2. The application will return to step 4.

## **Use Case 2: Login**

**Description:** User logs in with an existing account

**Actors:** User, Application System

**Pre-conditions:** The User has an existing account

**Post-conditions:** The User logs into the account successfully

**The flow of events:**

1. The application will prompt the User for his/her mobile number.
2. The User will input his/her mobile number.
3. The application will prompt the User for his/her password.
4. The User will input his/her password.
5. The application uses the included use case "Verification" to verify the User login details.
6. Upon successful verification, the User is logged into his account.

**Alternative flows:**

AF-S1: If any of the fields are blank.

1. The application will display "Error. There cannot be an empty field."
2. The application will return to step 1.

AF-S2: If no user account exists.

1. The application will display "Error. No user found."
2. The application will return to step 1.

AF-S4: If any of the fields are blank.

1. The application will display "Error. Incorrect password. Try again."
2. The application will return to step 1.

***Includes Use case "Verification"***

### **Use Case 3: Verification**

**Description:** The application verifies the login details of the User

**Actors:** User, Application System

**Pre-conditions:** The User has an existing account, with no changes to his/her mobile number

**Post-conditions:** The User login details are verified successfully

**The flow of events:**

1. The Application System retrieves the mobile number from the User input and searches its database for a valid mobile number.
2. The Application System retrieves the password from the User input and verifies in its database if it matches the password corresponding to the mobile number.
3. If the mobile number is valid, and the password corresponds, the Application System returns Boolean 'true' and displays the chosen user security question.
4. The application will prompt the User for the answer to the security question.
5. The User will key in the answer and the Application System will check if it corresponds.
6. If the answer is valid, the User is verified.

**Alternative flows:**

AF-S5: If the security answer is invalid

1. The Application System will display "Wrong answer. Please try again!".
2. The application will go back to step 3.

#### **Use Case 4: Reset Password**

**Description:** User resets password to an existing account

**Actors:** User, Application System

**Pre-conditions:** The User has an existing account, with no changes to his/her mobile number

**Post-conditions:** The User password is reset successfully

**The flow of events:**

1. The application will prompt the User for his mobile number.
2. The User will input his/her mobile number.
3. Upon mobile number verification, the application will prompt the User for 'new password' and 'password confirmation'.
4. The User will input the exact password for the 'new password' prompt and the 'password confirmation' prompt.
5. Once the 'new password' input and 'password confirmation' inputs are verified, the application will store the updated password in the system.
6. The password is reset for the User.

**Alternative flows:**

AF-S2: If no such user exists

1. The application will display "Error. No existing users!"
2. The application will return to step 1.

AF-S4: If the passwords do not meet the minimum length requirement.

1. The application will display "Error. Passwords must be at least 6 characters long."
2. The application will return to step 4.

AF-S4: If the passwords do not match.

1. The application will display "Passwords do not match. Please try again!"
2. The application will return to step 4.

### **Use Case 5: Help**

**Description:** User enquires for help regarding privacy concerns or FAQs.

**Actors:** User, Application System

**Pre-conditions:** The User is logged into his/her account

**Post-conditions:** The User is displayed options to enquire regarding privacy concerns or FAQs

**The flow of events:**

1. The application will use the included use case “Login” to verify the User.
2. The User clicks on the side panel button in the application.
3. The application displays the ‘Help’ option with subcategories to enquire about privacy concerns or FAQs with the extended use cases “Enquire Privacy Concerns” and “FAQ”.

*Extends Use case “Enquire Privacy Concerns” and “FAQ”*

### **Use Case 6: Enquire privacy concerns**

**Description:** User enquires about privacy concerns

**Actors:** User, Application System

**Pre-conditions:** The User is logged into his/her account

**Post-conditions:** The User is displayed with privacy protocols of the application

**The flow of events:**

1. The application will use the included use case “Login” to verify the User.
2. The User clicks on the option ‘Enquire privacy concerns’.
3. The application will display the privacy protocols of the application.

### **Use Case 7: FAQ**

**Description:** User enquires for FAQs.

**Actors:** User, Application System

**Pre-conditions:** The User is logged into his/her account

**Post-conditions:** The User is displayed with FAQs

**The flow of events:**

1. The application will use the included use case “Login” to verify the User.
2. The User clicks on the option ‘FAQs’.
3. The application will display the list of FAQs and their respective replies.

### **Use Case 8: Find Bicycle Parking Lot**

**Description:** The user is displayed a default number (5) of bicycle parking lots nearest to his/her specified destination

**Actors:** User, Application System, Parking Lot Database, Mapping Entity

**Pre-conditions:** The User is logged into his/her account, with GPS enabled

**Post-conditions:** Locations of bicycle parking lots nearest to his/her specified destination are displayed by the Mapping Entity

**The flow of events:**

1. The application will use the included use case "Login" to verify the User.
2. The User can filter bicycle parking lot coordinates based on shelter.
3. The Mapping Entity will retrieve the User's current coordinates using the included use case "Get User Coordinates" constantly in real-time and display his/her current location.
4. The User searches for a specific destination with the shelter filter.
5. The application will use the included use case "Search Location" to retrieve the coordinates of the specified destination.
6. The Mapping entity will then use the included use case "Search Location" to retrieve the nearest 5 bicycle parking lot coordinates.
7. The Mapping Entity will then display the default number (5) of bicycle parking lots using the included use case "Display Bicycle Parking Lots".
8. The User can select a bicycle parking lot.

***Includes Use case "Retrieve Bicycle Parking Lot Coordinates", "Get User Coordinates", "Login", "Display Bicycle Parking Lots", "Search Location"***

### **Use Case 9: Search Location**

**Description:** The User finds bicycle parking lots nearest to a specified destination

**Actors:** User, Application System, Parking Lot Database, Mapping Entity

**Pre-conditions:** The User is logged into his/her account, with GPS enabled

**Post-conditions:** Coordinates of bicycle parking lots are arranged in ascending order with respect to their distance to the user-specified destination, and returned to use case "Find Bicycle Parking Lot"

**The flow of events:**

1. The application will retrieve the coordinates of all filtered bicycle parking lots from the Parking Lot Database using the included use case "Retrieve Bicycle Lot Coordinates".
2. The User inputs the specified destination into the search bar.
3. The Mapping Entity will retrieve the coordinates of the destination and compare them with all the retrieved coordinates of the filtered bicycle parking lots.
4. The coordinates will be sorted in ascending order of distance from the specified destination and returned to the use case "Find Bicycle Parking Lot"

***Includes Use case "Get Destination Coordinates", "Retrieve Bicycle Lot Coordinates"***

### **Use Case 10: Add Destination**

**Description:** User can save commonly used destinations for a quick display of nearby bicycle parking lots

**Actors:** User, Application System, Mapping Entity

**Pre-conditions:** The User is logged into his/her account

**Post-conditions:** Destination coordinates are saved to the application, and the User will be able to find bicycle parking lots nearest to the saved destination

**The flow of events:**

1. The application will use the included use case “Login” to verify the User.
2. The User can navigate to the saved addresses tab under the side panel and search for multiple locations using the search bar on the application.
3. The User can select those desired destinations and the locations will be displayed in a list.
4. The User can set a location as ‘Home’.
5. The Mapping Entity will retrieve the destination coordinates and save them to the application system.
6. When the User wants to search for bicycle parking lots at the commonly used location, he/she can just click on the ‘Home’ button.
7. The Mapping Entity will then find the nearest bicycle parking lots using the included use case “Find Bicycle Parking Lot” and display them.

***Includes Use case “Login”, “Get Destination Coordinates”, “Find Bicycle Parking Lot”***



### **Use Case 11: Retrieve Bicycle Lot Coordinates**

**Description:** Retrieve all bicycle parking lot coordinates from the Parking Lot Database

**Actors:** Parking Lot Database, User

**Pre-conditions:** There are coordinates of existing bicycle parking lots within the Parking Lot Database

**Post-conditions:** The application can retrieve all bicycle parking lot coordinates for the User to carry out the use case “Find Bicycle Parking Lot”

**The flow of events:**

1. When the User executes the use case “Find Bicycle Parking Lot”, the application will request the data of all bicycle parking lots in the Parking Lot Database.
2. The Parking Lot Database will then return all data for each bicycle parking lot to the application, with the data consisting of lot coordinates, lot type, total capacity, and the presence of a shelter.
3. With the coordinates, the application can now compare them with the current coordinates of the User, or the destination coordinates, to complete the use case “Find Bicycle Parking Lot”.

### **Use Case 12: Get User Coordinates**

**Description:** Retrieve the current coordinates of the User using the Mapping Entity

**Actors:** Mapping Entity, User

**Pre-conditions:** GPS is enabled

**Post-conditions:** The application can retrieve the current coordinates of the User to carry out the use case “Find Bicycle Parking Lot”

**The flow of events:**

1. When the User clicks on the GPS button on the Map interface, the Mapping Entity will retrieve the user coordinates in real-time.
2. The Mapping Entity will then display his/her location on the map.
3. With the User coordinates, the application can update and display the live location of the User.

### **Use Case 13: Get destination coordinates**

**Description:** Retrieve the destination coordinates of the User using the Mapping Entity

**Actors:** Mapping Entity, User

**Pre-conditions:** Destination input is valid

**Post-conditions:** The application can retrieve all bicycle parking lot coordinates for the User to carry out the use case “Find Bicycle Parking Lot”

**The flow of events:**

1. When the User executes the use case “Find Bicycle Parking Lot”, with the search criteria based on the destination input, the application will request the coordinates of the destination.
2. The Mapping Entity will get the coordinates of the destination based on the User input and return them to the application.
3. With the destination coordinates, the application can now compare them with the coordinates of all bicycle parking lots, and complete the use case “Find Bicycle Parking Lot”.

### **Use Case 14: See Lot Details**

**Description:** The User can view attributes of a bicycle parking lot, like the presence of shelter, maximum capacity, and the lot name

**Actors:** Mapping Entity, User

**Pre-conditions:** Bicycle parking lots are displayed by the Mapping Entity

**Post-conditions:** The User can view all attributes of bicycle parking lots displayed by the Mapping Entity, along with corresponding reports on it by other users

**The flow of events:**

1. After the location of the nearest bicycle parking lots are displayed using the Mapping Entity after executing the use case “Find Bicycle Parking Lot”, the User selects a bicycle parking lot.
2. The User clicks on the location marker.
3. The application will display the attributes of the selected bicycle parking lot.

### **Use Case 15: Display Bicycle Parking Lots**

**Description:** Display nearby bicycle parking lots using the Mapping Entity

**Actors:** Mapping Entity, User

**Pre-conditions:** Coordinates of nearby bicycle parking lots are retrieved from the Parking Lot Database

**Post-conditions:** The locations of nearby bicycle parking lots are displayed by Mapping Entity

**The flow of events:**

1. After the User executes the use case “Find Bicycle Parking Lot”, the application would have obtained the coordinates of the nearest bicycle parking lots.
2. The coordinates of the 5 bicycle parking lots will be passed to the Mapping Entity, and displayed by default unless stated differently by the user.
3. The User will be able to view the location of the nearest bicycle parking lots via the Mapping Entity.

**Alternative flows:**

AF-S3: If the User clicks on the “See more” option to view more bicycle parking lots

1. The application will display more locations of nearby bicycle parking lots through the Mapping Entity using the extended use case “See More”.

***Extends use case “See More Lots”, “See Lot Details”, “Directions”***

### **Use Case 17: Directions**

**Description:** The User can view directions to the selected bicycle parking lot

**Actors:** Mapping Entity, User

**Pre-conditions:** Current coordinates of the User and bicycle parking lot coordinates are available to the Mapping Entity

**Post-conditions:** The User can view directions to the selected bicycle parking lot from the current location via the Mapping Entity

**The flow of events:**

1. After the locations of the nearest 5(default) bicycle parking lots are displayed using the Mapping Entity after executing the use case “Find Bicycle Parking Lot”, the User selects a bicycle parking lot.
2. The User clicks on the “Directions” option.
3. The Mapping Entity will show the directions from the current location of the User to the location of the selected bicycle parking lot.

### **Use Case 16: See More Lots**

**Description:** The User can expand the view of the nearest bicycle parking lots from the nearest 5 to the next nearest 5 bicycle parking lots

**Actors:** Mapping Entity, User

**Pre-conditions:** Coordinates of nearby bicycle parking lots are retrieved from the Parking Lot Database

**Post-conditions:** The additional locations of nearby bicycle parking lots are displayed by the Mapping Entity

**The flow of events:**

1. After the location of the nearest 5 bicycle parking lots is displayed using the Mapping Entity after executing the use case “Find Bicycle Parking Lot”, the User clicks on the “See more” option.
2. The application will pass the coordinates of the next 5 nearest bicycle parking lots to the Mapping Entity.
3. The Mapping Entity will display the locations of the additional 5 bicycle parking lots.
4. The User will be able to view the location of the nearest 5 bicycle parking lots via the Mapping Entity.

**Alternative flows:**

AF-S4: If the User decides to click on the “See more” option to view more bicycle parking lots

1. The application will return to Step 2 and then Step 5.
2. If all bicycle parking lots are displayed, clicking on the “See more” option will not display additional bicycle parking lots.

### **Use Case 17: Share Ride**

**Description:** The user can share his ride destination, current location, and nearby bicycle parking lots with others via a hyperlink

**Actors:** User, Application System, Parking Lot Database, Mapping Entity

**Pre-conditions:** The User is logged into his/her account, with GPS enabled

**Post-conditions:** The location of the User, together with his ride destination will be shared with other users when they access the hyperlink

**The flow of events:**

1. The application will use the included use case "Login" to verify the User.
2. The User can find a bicycle parking lot using the included use case "Find Bicycle Parking Lot"
3. The location of the User, along with the ride destination will be updated and displayed using the included use case "Find Bicycle Parking Lot".
4. The locations of the bicycle parking lots will be displayed by the Mapping Entity.
5. The user then clicks on the "Share Ride" option.
6. The application will generate a link to the map.
7. The User sends the link to other users.
8. Other users are now able to view the current location and the ride destination of the User by accessing the link.

***Includes Use case "Login", "Find Bicycle Parking Lot".***

### **Use Case 18: Group ride**

**Description:** The host User can invite other users on a group ride via a hyperlink that shares the destination, nearby bicycle parking lots, and current locations of all users in the group

**Actors:** User, Application System, Parking Lot Database, Mapping Entity

**Pre-conditions:** The host User is logged into their accounts, with GPS enabled

**Post-conditions:** Ride destination, nearby bicycle parking lots and the current location of all users within the party are shared amongst each other

**The flow of events:**

1. The application will use the included use case “Login” to verify the host User.
2. The host User can find a bicycle parking lot using the included use case “Find Bicycle Parking Lot”
3. The location of the host User, along with the ride destination will be updated and displayed by the Mapping Entity using the included use case “Find Bicycle Parking Lot”.
4. The locations of the bicycle parking lots will be displayed using the Mapping Entity.
5. The host User then clicks on the “Group Ride” option.
6. The application will generate a link to the map.
7. The host User sends the link to other users.
8. After other users access the link, the GPS will retrieve the current coordinates of all other users and display them using the Mapping Entity.
9. All users within the group can view the current locations of everyone and the ride destination, together with bicycle parking lots nearest to the host User.

***Includes Use case “Find Bicycle Parking Lot”, “Login”.***