Use Cases

for

TwoTyred Web Application

**Version 1.0 approved**

**Prepared by**

**Ang Kai Jun, Ng Li Lin Evonne, Liu Liwen,**

**Chay Hui Xiang, Ivan Loke Zhi Hao, Chang Dao Zheng**

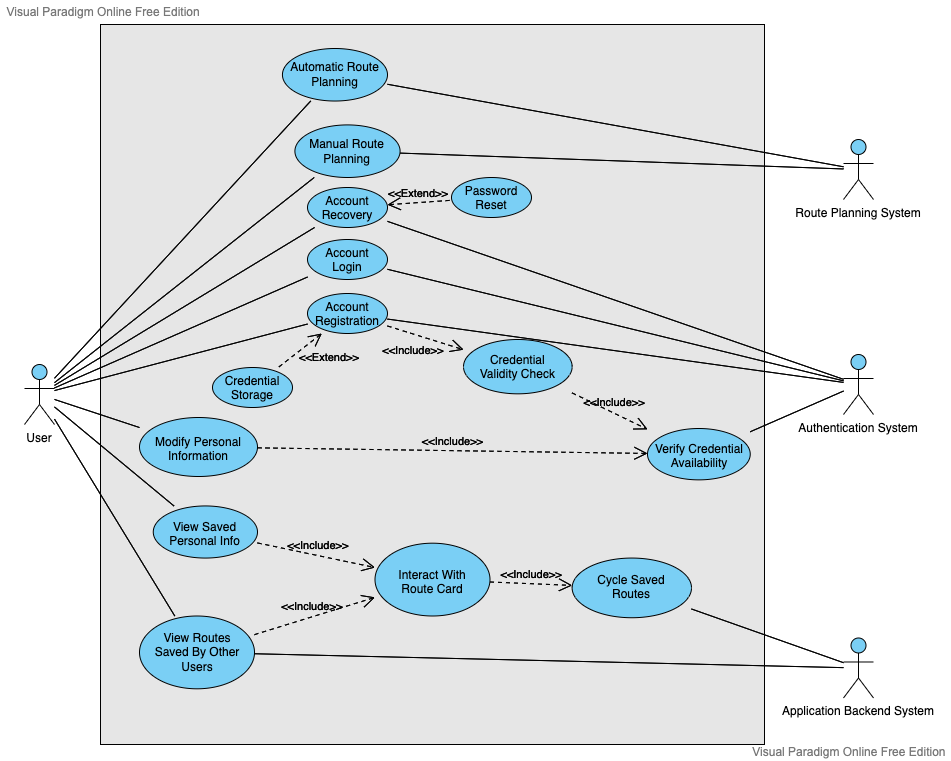
**SC2006 Software Engineering, REP Lab Group, Team Fantastic 4**

**6th November 2022**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| All members | 6/11/2022 | Populate Content | 1.0 |
|  |  |  |  |

# Use Case Diagram



# 

# Use Case Hierarchy

|  |  |  |
| --- | --- | --- |
| **Group No.** | **Group Name** | **Use Cases** |
| 1 | Authentication Use Cases | 1.1 Account Registration |
| 1.2 Credential Storage |
| 1.3 Credential Validity Check |
| 1.4 Verify Credential Availability |
| 1.5 Account Login |
| 1.6 Account Recovery |
| 1.7 Password Reset |
| 2 | Social Media Use Cases | 2.1 Modify Personal Information |
| 2.2 View Saved Personal Information |
| 2.3 View Routes Saved by Others |
| 2.4 Interact With Route Cards |
| 3 | Route Planning Use Cases | 3.1 Manual Route Planning |
| 3.2 Automatic Route Planning |
| 3.3 Cycle Saved Route |

# 

# 

# Use Case Descriptions

**Group 1 (Authentication Use Cases):**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.1 | | |
| Use Case Name: | Account Registration | | |
| Created By: | Chay Hui Xiang | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 18 October |

|  |  |
| --- | --- |
| Actor: | User, Authentication System |
| Description: | Registers an account for the user |
| Preconditions: | – |
| Postconditions: | 1. The user registers an account with his/her own set of input credentials |
| Priority: | Medium |
| Frequency of Use: | Low |
| Flow of Events: | 1. The frontend prompts the user for their account username, email address and password. 2. The user inputs their desired account username, email address and password. 3. The user submits their account credentials for registration via the submit button. 4. After the user has submitted their desired account credentials, the system uses the included use case Credential Validity Check to check the input credentials. 5. The system uses the included use case Credential Storage and stores all sign-up credentials in the database. |
| Alternative Flows: | – |
| Exceptions: | 1.1.EX.1: If the user is facing network issues   1. The frontend displays the message “Network error”. |
| Includes: | Credential Validity Check, Credential Storage |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.2 | | |
| Use Case Name: | Credential Storage | | |
| Created By: | Chay Hui Xiang | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 18 October |

|  |  |
| --- | --- |
| Actor: | Authentication System |
| Description: | Stores username, password and email into the database |
| Preconditions: | 1. Database has sufficient storage available for credential storage. 2. User credentials must be verified and valid. |
| Postconditions: | 1. New document added into the database with new user credentials. |
| Priority: | Medium |
| Frequency of Use: | Low |
| Flow of Events: | 1. Authentication System receives credential input from user. 2. Authentication System stores the username, password and email into the database. |
| Alternative Flows: | – |
| Exceptions: | 1.2.EX.1: If the authentication system is unable to connect to the database   1. The frontend displays the message “Network error”. 2. The frontend redirects the user back to the account registration page. |
| Includes: | – |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.3 | | |
| Use Case Name: | Credential Validity Check | | |
| Created By: | Chay Hui Xiang | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 18 October |

|  |  |
| --- | --- |
| Actor: | Authentication System |
| Description: | Checks the validity of the credentials keyed in by the user |
| Preconditions: | 1. The user must have inputted a set of user credentials. |
| Postconditions: | 1. The authentication system checks the validity of the user’s credentials. |
| Priority: | Medium |
| Frequency of Use: | Low |
| Flow of Events: | 1. Authentication System receives credential input from user. 2. Authentication System uses the included use case Verify Credential Availability to verify that the username and email input from the user is available. 3. Authentication System checks if the password input meets the password requirement (at least 8 characters, at least 1 uppercase and lowercase letter each, at least 1 number, at least 1 symbol). 4. Authentication System checks if the input from the password field and confirm password field are exactly the same. |
| Alternative Flows: | – |
| Exceptions: | 1.3.EX.1: If the input password does not meet the password length requirement   1. The frontend displays the message “Password length too short.”   1.3.EX.2: If the input password does not meet the password lowercase requirement   1. The frontend displays the message “Password does not contain lowercase letters.”   1.3.EX.3: If the input password does not meet the password uppercase requirement   1. The frontend displays the message “Password does not contain uppercase letters.”   1.3.EX.4: If the input password does not meet the password number requirement   1. The frontend displays the message “Password does not contain a number.”   1.3.EX.5: If the input password does not meet the password symbol requirement   1. The frontend displays the message “Password does not contain a symbol.”   1.3.EX.6: If the input from the password and confirm password fields are not the same   1. The frontend displays the message “Passwords do not match.” |
| Includes: | Verify Credential Availability |
| Special Requirements: | - |
| Assumptions: | - |
| Notes and Issues: | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.4 | | |
| Use Case Name: | Verify Credential Availability | | |
| Created By: | Chay Hui Xiang | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 18 October |

|  |  |
| --- | --- |
| Actor: | Authentication System |
| Description: | Checks that the username and email input from the user is available. |
| Preconditions: | 1. The user must have submitted a username and email input. |
| Postconditions: | 1. The authentication system must have checked the availability of the username and email input. |
| Priority: | Medium |
| Frequency of Use: | Low |
| Flow of Events: | 1. Authentication System receives username and email input from the user. 2. Authentication System cross-checks the username and email input with the database to ensure that both inputs are available. |
| Alternative Flows: | – |
| Exceptions: | 1.4.EX.1: If the username input is not available   1. The frontend displays the message “Username has already been taken.”   1.4.EX.2: If the email input is not available   1. The frontend displays the message “The email has already been taken.” |
| Includes: | – |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.5 | | |
| Use Case Name: | Account Login | | |
| Created By: | Chay Hui Xiang | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 18 October |

|  |  |
| --- | --- |
| Actor: | User, Authentication System |
| Description: | Logs the user in to a registered account |
| Preconditions: | – |
| Postconditions: | The user must be logged in to a registered account. |
| Priority: | Medium |
| Frequency of Use: | High |
| Flow of Events: | 1. The system prompts the user for their account username and password. 2. The user enters his/her account username and password into the respective fields on the Login page. 3. The system cross-checks the input credentials with those from their database. 4. The frontend redirects the user back to the dashboard page. |
| Alternative Flows: | 1.5.AC.1: If the input credentials are wrong   1. The frontend displays the message “Wrong username or password.” 2. The frontend returns to step 1.   1.5.AC.2: If the input credentials are wrong for 5 times   1. The frontend displays the message “You have been locked out of your account.” 2. The authentication system locks the user out of their account for 10 minutes. 3. The authentication system sends an email to the user’s email to inform them that their account has been temporarily locked for 10 minutes, and recommend them to change their password if they were not the ones trying to login to the account. 4. The frontend returns to step 1 |
| Exceptions: | 1.5.EX.1: If the user is facing network issues   1. The frontend displays the message “Network error”. |
| Includes: | – |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.6 | | |
| Use Case Name: | Account Recovery | | |
| Created By: | Chay Hui Xiang | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 18 October |

|  |  |
| --- | --- |
| Actor: | User, Authentication System |
| Description: | Recovers the user’s account in case he/she has forgotten his/her password |
| Preconditions: | – |
| Postconditions: | 1. The user’s account must now be redirected to the password reset screen. |
| Priority: | Medium |
| Frequency of Use: | Low |
| Flow of Events: | 1. The user clicks on the “Forget Password” option. 2. The frontend prompts the user for their email account. 3. The user inputs their email account. 4. The system cross-checks the inputted email with the database to ensure that an account with the associated email exists. 5. The system sends an email to the email address with an email containing the password reset link. 6. The user clicks on the password reset link, which redirects them to the account recovery page. 7. The system uses the included use case Password Reset to then reset the password for the user |
| Alternative Flows: | 1.6.AC.1: If there are no accounts associated with the inputted email   1. The frontend displays the message “No accounts found.” 2. The frontend returns to step 2. |
| Exceptions: | 1.6.EX.1: If the user is facing network issues   1. The frontend displays the message “Network error”. |
| Includes: | Password Reset |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.7 | | |
| Use Case Name: | Password Reset | | |
| Created By: | Chay Hui Xiang | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 18 October |

|  |  |
| --- | --- |
| Actor: | User, Authentication System |
| Description: | Reset password for users who have forgotten their account credentials |
| Preconditions: | User is on the account recovery page |
| Postconditions: | 1. The user’s account must now be reset. |
| Priority: | Medium |
| Frequency of Use: | Low |
| Flow of Events: | 1. The system prompts the user for a new password. 2. The user submits their new password for account recovery via the submit button. 3. After the user has submitted their new password, the system uses the included use case Credential Validity Check to check the validity of the new password. 4. The frontend returns the message “Password changed”. |
| Alternative Flows: | – |
| Exceptions: | 1.7.EX.1: If the user is facing network issues   1. The frontend displays the message “Network error”. |
| Includes: | Credential Validity Check |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

**Group 2 (Social Media Use Cases):**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.1 | | |
| Use Case Name: | Modify Personal Information | | |
| Created By: | Chang Dao Zheng | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 18 October |

|  |  |
| --- | --- |
| Actor: | User, Authentication System, Application Backend System |
| Description: | Allow user to modify their username and profile picture |
| Preconditions: | 1. The user must be logged in to modify username and profile picture. 2. The application must have access to the user’s local storage or local camera device (if any). |
| Postconditions: | 1. The new profile picture must be saved in the Application Backend System. 2. The new username must be saved in the Authentication System. 3. The new username and profile picture must be displayed in the user profile page. |
| Priority: | Low |
| Frequency of Use: | Low |
| Flow of Events: | 1. The user will request to edit their personal profile. 2. The user can choose to upload a new photo from their local storage or to take a new picture with the local camera device. 3. The Application Backend System will store the new profile picture in the database. 4. The user can also choose to give a new username. 5. When the user confirms the modification of username, he triggers the Verify Credential Availability use case to check for duplicates of the username. 6. If there are no duplicated usernames, the Authentication System will update the user’s username in the database. |
| Alternative Flows: | 2.1.AC.1: If the username is already in use   1. The Authentication System will reject the updating of the new username. 2. The Authentication System will send an error message to the application front-end. 3. The application returns to step 2. |
| Exceptions: | 2.1.EX.1: If the user is facing network issues   1. The Authentication System will reject the updating of the new username. |
| Includes: | Verify Credential Availability |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.2 | | |
| Use Case Name: | View Saved Personal Information | | |
| Created By: | Chang Dao Zheng | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 18 October |

|  |  |
| --- | --- |
| Actor: | User, Application Backend System |
| Description: | Users can view their profile picture, username, past routes and routes that they selected as “Favourites” |
| Preconditions: | 1. The user must be logged in. |
| Postconditions: | 1. The user must be able to view information in the application. |
| Priority: | Medium |
| Frequency of Use: | High |
| Flow of Events: | 1. The user should be able to see their latest username and profile picture when the click on the application’s personal profile tab. 2. The user should also be on the “Past Routes” tab and should be able to see a list of past rides in the form of multiple route cards. This should come with details such as ride distance and number of likes for each route. 3. If the user clicks on any part of the route card, he/she will trigger the included Interact With Route Card use case. 4. If the user clicks on the “Favourites” tab, the Application Backend System will show a list of routes that they have previously favourited on the dashboard page |
| Alternative Flows: | 2.2.AC.1: If there are no previous rides   1. The application will display an empty profile dashboard with no route cards. |
| Exceptions: | 2.2.EX.1: If the user is facing network issues   1. The frontend displays the message “Network error”. |
| Includes: | Interact With Route Card |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.3 | | |
| Use Case Name: | View routes saved by other users | | |
| Created By: | Chang Dao Zheng | Last Updated By: | Chang Dao Zheng |
| Date Created: | 25 August | Date Last Updated: | 30 August |

|  |  |
| --- | --- |
| Actor: | User, Application Backend System |
| Description: | Users can interact with the routes saved by other users, including riding the route saved and liking the route |
| Preconditions: | 1. The user must be logged in. |
| Postconditions: | 1. The user must be able to interact with other user’s saved routes when they click on the application’s dashboard tab. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. The user should be able to see a list of routes saved by other users, in the form of a route card, when they select the application’s dashboard tab. The route cards will be sorted based on the greatest number of likes by default. 2. If the user clicks on any part of the route card, he/she will trigger the included Interact With Route Card use case. 3. If the user chooses to click on the dropdown menu and sorts by date, the Application Backend System will return a list of route cards, sorted by their recency, to the Application’s frontend. |
| Alternative Flows: | – |
| Exceptions: | 2.3.EX.1: If the user is facing network issues   1. The frontend displays the message “Network error”. |
| Includes: | Interact With Route Card |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2.4 | | |
| Use Case Name: | Interact with route card | | |
| Created By: | Chang Dao Zheng | Last Updated By: | Chang Dao Zheng |
| Date Created: | 25 August | Date Last Updated: | 18 October |

|  |  |
| --- | --- |
| Actor: | User, Application Backend System |
| Description: | Users can interact with the route card to like and favourite the route card, or be redirected to view more details about the route and cycle the route. |
| Preconditions: | 1. The user must be logged in. 2. The user must be on a page where he can view a route card on the user interface |
| Postconditions: | 1. The user successfully interacts with a route card. |
| Priority: | Medium |
| Frequency of Use: | High |
| Flow of Events: | 1. The user should be able to see a route card. Each of these route cards should have a like button (heart), a like counter, a favourite button (star). 2. If the user clicks on the like button, the Application Backend System should record a like for the route in the database and increase the like counter accordingly on the application frontend. 3. If the user clicks on the favourite button, the Application Backend System should store the route into the user’s database and subsequently display the route in the user’s “Favourites” tab in his/her User Profile page. 4. If the users were to click on any other parts of the route card, the application will redirect them to the route description page, where they can view more details about the route. 5. The users will be able to select a “Cycle Route” button, which will trigger the Cycle Saved Route use case. 6. The Backend Application System will record the ride in the user’s database and subsequently display the route in the user’s “Past Routes” tab in his/her User Profile page. |
| Alternative Flows: | – |
| Exceptions: | 2.4.EX.1: If the user is facing network issues   1. The frontend displays the message “Network error”. |
| Includes: | – |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

**Group 3 (Route Planning Use Cases):**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.1 | | |
| Use Case Name: | Manual Route Planning | | |
| Created By: | Chay Hui Xiang | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 30 August |

|  |  |
| --- | --- |
| Actor: | User, Authentication System |
| Description: | Plan routes based on start, intermediate and destination points input by the user |
| Preconditions: | 1. The user must be logged in to a registered account. 2. Database has sufficient storage available for route storage. |
| Postconditions: | 1. A route must be recommended based on the user’s start, end and intermediate points input. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. The frontend prompts the user for their start point, end point and intermediate points along the route. 2. The user selects their route points via map markers or address input. 3. The route planning system will determine the optimal route, PM2.5 index and weather data by querying the OneMap API and weather API. 4. The frontend will display the proposed route, along with the route distance, and the latest PM2.5 index and weather data. 5. The user will choose to “Cycle Route” or “Generate another route”. 6. In the case that the user chooses to “Cycle Route”, the route will be saved under the user’s profile. 7. In the case that the user chooses to “Generate another route”, the frontend will redirect the user back to step 1. |
| Alternative Flows: | – |
| Exceptions: | 3.1.EX.1: If the user is facing network issues   1. The frontend displays the message “Network error”. |
| Includes: | – |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.2 | | |
| Use Case Name: | Automatic Route Planning | | |
| Created By: | Chay Hui Xiang | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 30 August |

|  |  |
| --- | --- |
| Actor: | User, Authentication System |
| Description: | Plan routes based on start point and cycling distance input by the user |
| Preconditions: | 1. The user must be logged in to a registered account. 2. Database has sufficient storage available for route storage. |
| Postconditions: | 1. A route must be recommended based on the user’s start point and cycling distance inputs. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. The frontend prompts the user for their start point and distance 2. The user selects their start point via map markers or address input. 3. The user inputs the distance into a field on the frontend 4. The route planning system will determine the optimal route, PM2.5 index and weather data by querying the I’mFeelingLucky backend and Data.gov Weather API. 5. The frontend will display the proposed route, along with the route distance, and the latest PM2.5 index and weather data. 6. The user will choose to “Cycle Route” or “Generate another route”. 7. In the case that the user chooses to “Cycle Route”, the route will be saved under the user’s profile. 8. In the case that the user chooses to “Generate another route”, the frontend will redirect the user back to step 1. |
| Alternative Flows: | – |
| Exceptions: | 3.2.EX.1: If the user is facing network issues   1. The frontend displays the message “Network error”. |
| Includes: | – |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3.3 | | |
| Use Case Name: | Cycle saved routes | | |
| Created By: | Chang Dao Zheng | Last Updated By: | Chang Dao Zheng |
| Date Created: | 25 August | Date Last Updated: | 19 October |

|  |  |
| --- | --- |
| Actor: | User, Application Backend System |
| Description: | Users can ride routes that have been created by themselves or other users. |
| Preconditions: | 1. The user must be logged in 2. The user must be on a page where he/she is able to access a route’s description and click on the “Cycle Route” button. |
| Postconditions: | 1. The ride must be saved under the “Past Routes” tab |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. When the user clicks on the view more link, he will be led to a page with more details about the route. A “Cycle Route” option will be provided as well. 2. When the user clicks on the “Cycle Route” option, the Application Backend System will record the ride in the user’s database and subsequently display the route under the Past Rides tab in the User Profile page. |
| Alternative Flows: | – |
| Exceptions: | 3.3.EX.1: If the user is facing network issues   1. The frontend displays the message “Network error”. |
| Includes: | – |
| Special Requirements: | – |
| Assumptions: | – |
| Notes and Issues: | – |