**Functional Requirements Elicitation:**

1. The website must consist of a User Authentication Feature
   1. The user must be able to sign up for a new account
      1. To sign up for a new account, the user must provide their username
      2. To sign up for a new account, the user must provide their email address
      3. To sign up for a new account, the user must provide their password
      4. To sign up for a new account, the user must provide the same password in the “confirm password” field
         1. The system must be able to check if the password in both the “password” and “confirm password” field is the same.
         2. If the passwords entered into the “password” and “confirm password” fields are different, the system must prompt an error.
         3. The user must only be able to sign up when both passwords inputted are identical.
   2. The user must be able to log in to existing account
      1. To log into their account, the user must input their correct email and password
      2. If the email or password provided is wrong, the system must prompt an error
   3. The user must be able to change their password when they forget their password
      1. The user must be able to input their email to receive an email containing password reset link
      2. The user must be able to change their password after clicking on the password reset link, which can be found in an email that the system sent
      3. The user must be able to key in their new password
         1. The system must be able to validate if the new password is the same as the current password. If they are the same, the system must prompt an error
         2. The system must be able to check if the user keyed in the same password in the “password” field and the “confirm password” field
      4. The user must be able to log into their account with the new password
2. **The website must consist of a User Profile Feature**
   1. The user must be able to upload a picture as their profile picture
      1. The user must be able to change their picture
   2. The user must be able to see their username
      1. The user must be able to change their username
   3. The user must be able to see the total distance cycled
   4. The user must be able to see the total amount of time spent cycling
   5. The user must be able to see all their previous routes
      1. The user must be able to create a new route that is identical to any previous route selected.
   6. The user must be able to add their previous routes to a “favourites” list
      1. The user must be able to only view favourite routes using a filter
3. **The website must consist of a Route Planning Feature**
   1. The website provides capabilities to plot cycling routes only.
   2. To use the Route Planning Feature, users must input location information.
      1. The user must input the start destination
      2. The user must input an end destination
      3. The user is given the option to add extra destinations along the route.
   3. After the user provides all the location information, the system will return the proposed route.
      1. The system must make use of the OneMap API to provide the proposed routes.
      2. The proposed route must be represented on a map
      3. There must be a clear indication of the whole cycling route, including the start and the end point.
      4. The route distance must be indicated.
      5. The PM2.5 index for the route must be indicated.
      6. The weather data for the route must be indicated.
   4. Based on the proposed route returned, the user must be able to choose to “Generate another route” or “Cycle route”.
      1. If the user chooses “Cycle route”, the user is assumed to have cycled the route and the route must be saved into the database.
      2. If the user chooses “Generate another route”, the user must be able to input a new set of location information to generate a new route.
4. **The website must consist of a Dashboard Feature**
   1. The system must have a database for all routes planned by all its users
   2. The system must display cycling routes by all users which have been plotted by the website
      1. The system must display the start destination
      2. The system must display the end destination
      3. The system must display the date the route was plotted
      4. The system must display the distance of the route
      5. The system must display the weather conditions
      6. The system must display the total number of “likes”
      7. The user must be able to create a new route that is identical to any route shown on the dashboard.
      8. The user must be able to "favourite” any route on the Dashboard, and add it to their favourite list.
   3. The user must be able to “like” routes
   4. The user must be able to sort routes
      1. The user must be able to sort routes based on total number of “likes”
      2. The user must be able to sort routes based on how recent the route was plotted
   5. The user must be able to change the view of the dashboard
      1. The user must be able to change the view to card view
      2. The user must be able to change the view to large tiles
      3. The user must be able to change the view to small tiles
   6. The system must be able to load more routes when the user scrolls down
   7. The user must be able to access all other features of the app
      1. The system must include a menu to other features of the app
         1. The system must provide a path to user profile
         2. The system must provide a path to route planning
         3. The system must provide a path to the dashboard
      2. The user must be able to access these features when clicking on the item in the menu

**Non-functional Requirements Elicitation**

1. User Authentication Features
   1. Sign Up
      1. The system must store all sign-up credentials into the database (i.e. username, email and password)
      2. The system must be able to check if the email address provided has already been used.
         1. If the mail address has already been used, the system must prompt the user to input a new email address.
      3. The system must be able to check if the username chosen has already been used.
         1. If the username has already been used, the system must prompt the user to input a new username
      4. The system must be able to check if the email input has already been used.
         1. If the email has already been used, the system must prompt the user to input a new email
      5. The system must be able to check if the password meets the password requirement
         1. All passwords must consist of at least 8 characters
         2. All passwords must consist of at least 1 uppercase letter
         3. All passwords must consist of at least 1 lowercase letter
         4. All passwords must consist of at least 1 number
         5. All passwords must consist of at least 1 special letter (eg ! @ # ?)
   2. Log in
      1. The system must temporarily lock the account for 10 mins if the credentials have been entered wrongly 5 times.
      2. The system must send an email to the user’s email to inform them that their account is temporarily locked for 10mins, and recommend them to change their password if they were not the ones trying to log-in to the account.
   3. Change Password
      1. The system must send a notification email that the password was changed recently within 10 minutes of password reset.
2. **User Profile Feature**
   1. The system must be able to have access to the user’s gallery
   2. The system must upload the user’s selected route to their profile within 5 minutes.
   3. The system must store all cycling routes in the database
3. **Route Planning Feature**
   1. The system must have access to the map of Singapore.
   2. The system must be able to allow users to indicate their start, intermediate and end points.
      1. The system must allow users to either search for a location in a search box, or indicate a location by clicking on the specific location on the map.
      2. Once all the location information is provided, the system must be able to provide the proposed route within 1 minute.
      3. The information (selected points, route, pm2.5 and weather data) provided on the proposed routes must be accurate at all times
   3. Once the user chooses “Cycle route”, the route must be reflected in the User Profile and Dashboard.
4. **Dashboard Feature**
   1. The system must be able to increase or decrease the size of the image of routes planned according to the user’s viewing preference
      1. The system must be able to change the size within 5 seconds of change of viewing preference
   2. The system must be able to load more routes within 5 seconds of a user scrolling

**Data Dictionary**

|  |  |
| --- | --- |
| Term | Definition |
| Route | A way or course taken from a starting point to an ending point. |
| Destination | The end point a user wants to go to. |
| Weather Conditions | The temperature and prediction of rain of a certain date and time. |
| API | Application Programming Interface. |
| Dashboard | A user interface page that will display the routes created by other users. |
| User Profile | A display of user’s information and cycling history. |
| User Authentication | A process that allows user to sign up, log-in, choose username and set password. |
| Route Card | A user interface component in the User Profile and Dashboard where users can view the summary of a route |
| Database | Data storage for all user and route information. |

**Use Case Models**

Use Case Diagram

Diagram

Description automatically generated

Use Case Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 001 | | |
| Use Case Name: | Credential Storage | | |
| Created By: | Chay Hui Xiang | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 30 August |

|  |  |
| --- | --- |
| 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. |
| 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: | EX1. 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 login page. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 002 | | |
| Use Case Name: | Credential Validity Check | | |
| Created By: | Chay Hui Xiang | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | Date Last Updated: | 30 August |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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. | | |
| 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: | | EX1: If the input password does not meet the password requirement   1. The frontend displays the message “Password does not meet requirements. Please try again.”   EX2: If the input from the password and confirm password fields are not the same   1. The frontend displays the message “Passwords do not match. Please try again.” | | |
| Use Case ID: | 003 | | | |
| Use Case Name: | Verify Credential Availability | | | |
| Created By: | Chay Hui Xiang | | Last Updated By: | Chay Hui Xiang |
| Date Created: | 25 August | | Date Last Updated: | 30 August |

|  |  |
| --- | --- |
| 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. |
| 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: | EX1: If the username input is not available   1. The frontend displays the message “Username has already been taken. Please try again.”   EX1: If the email input is not available   1. The frontend displays the message “ The email has already been taken. Please try again.” |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 004 | | |
| Use Case Name: | Account Registration | | |
| 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: | Registers an account for the user |
| Preconditions: |  |
| Postconditions: | 1. The user registers an account with his/her own set of input credentials |
| 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: | EX1. If the user is facing network issues   1. The frontend displays the message “Network error”. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 005 | | |
| Use Case Name: | Account Recovery | | |
| 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: | 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. |
| 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: | AF-S4: If there are no accounts associated with the inputted email   1. The frontend displays the message “No account found. Please try again.” 2. The frontend returns to step 2. |
| Exceptions: | EX1. If the user is facing network issues   1. The frontend displays the message “Network error”. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 006 | | |
| Use Case Name: | Password Reset | | |
| 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: | Reset password for users who have forgotten their account credentials |
| Preconditions: |  |
| Postconditions: | 1. The user’s account must now be reset. |
| 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 Reset”. |
| Alternative Flows: |  |
| Exceptions: | EX1. If the user is facing network issues   1. The frontend displays the message “Network error”. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 007 | | |
| Use Case Name: | Account Login | | |
| 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: | Logs the user in to a registered account |
| Preconditions: |  |
| Postconditions: | 1. The user must be logged in to a registered account. |
| 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 Log In page. 3. The system cross-checks the input credentials with those from their database. 4. The frontend redirects the user back to the home page. |
| Alternative Flows: | AF-S3: If the input credentials are wrong   1. The frontend displays the message “Wrong username or password. Please try again.” 2. The frontend returns to step 1.   AF-S3: If the input credentials are wrong for 5 times   1. The frontend displays the message “You have been locked out of your account. Please try again later.” 2. The authentication system locks the user out of their account for 10 minutes. 3. The authentication system locks the user out of their account for 10 minutes. 4. 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. 5. The frontend returns to step 1 |
| Exceptions: | EX1. If the user is facing network issues   1. The frontend displays the message “Network error”. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 008 | | |
| Use Case Name: | 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. |
| 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, 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: | EX1. If the user is facing network issues   1. The frontend displays the message “Network error”. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 009 | | |
| Use Case Name: | Modify Personal Information | | |
| Created By: | Chang Dao Zheng | Last Updated By: | Chang Dao Zheng |
| Date Created: | 25 August | Date Last Updated: | 30 August |

|  |  |
| --- | --- |
| 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. |
| 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 Username 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: | AF-5: 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: | EX1. If the user is facing network issues   1. The Authentication System will reject the updating of the new username. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 010 | | |
| Use Case Name: | View Saved Personal Information | | |
| 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 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. |
| 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 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 “Saved Routes” tab, the Application Backend System will show a list of routes that they have previously created on the application front-end |
| Alternative Flows: | AF-2: If there are no previous rides   1. The application will show a “No Previous Runs” message |
| Exceptions: | EX1. If the user is facing network issues   1. The frontend displays the message “Network error”. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 011 | | |
| 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. |
| 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 routes 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: | EX1. If the user is facing network issues   1. The frontend displays the message “Network error”. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 012 | | |
| 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: | 30 August |

|  |  |
| --- | --- |
| 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. |
| 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 favourites 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 add the route into 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 a separate 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 database and this will be reflected in the user’s “Past Routes” tab in his/her User Profile page. |
| Alternative Flows: |  |
| Exceptions: | EX1. If the user is facing network issues   1. The frontend displays the message “Network error”. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 013 | | |
| Use Case Name: | Cycle saved routes | | |
| 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 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 |
| 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 data base and show the ride in under the Past Rides tab in the User Profile page. |
| Alternative Flows: |  |
| Exceptions: | EX1. If the user is facing network issues   1. The frontend displays the message “Network error”. |

**User Interface Mock-ups**

Figma Link: <https://www.figma.com/file/0lfndaaCnz2mNcinT6pdc0/UI-Mock-ups?node-id=0%3A1>

User Authentication

1. Sign In Page

Graphical user interface

Description automatically generated

1. Account Registration Page

Graphical user interface

Description automatically generated

1. Reset Password Page

Graphical user interface, application

Description automatically generated

1. Sign Up Successful Page

Graphical user interface

Description automatically generated

Route Selection

1. Start Point Selection

Graphical user interface, application, map

Description automatically generated

1. Point Selection

Graphical user interface, application, map

Description automatically generated

Graphical user interface, application, map

Description automatically generated

Graphical user interface, application, map

Description automatically generated

1. Create Cycling Route

Graphical user interface, application

Description automatically generated

1. Cycling Route Description

Graphical user interface, application, map

Description automatically generated

* 1. Note: “Route successfully saved!” alert will only show after the user clicks on the “Cycle Route” link

User Profile

1. User Profile Page

Graphical user interface, website

Description automatically generated

Dashboard

1. Dashboard Page

Graphical user interface, application, website

Description automatically generated