

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

SC2006: Software Engineering

Lab 1 Deliverables

AY24/25 | SEM 2 | SCSC | Group 3

Name	Matriculation Number
Harikrishnan Vinod	U2321114H
Bryan Goh Wei Hao	U2221356B
Ethan Yew Yi Teng	U2321995C
Lim En Jia	U2320279L
Chin Hui Qi Cheryl	U2321555A

Table of Contents

Table of Contents.....	2
Introduction/Mission statement.....	3
Documentation of Functional and Non-Functional Requirements.....	3
1. Functional Requirements.....	3
2. Non-Functional Requirements.....	8
2.1. Performance.....	8
2.2. Scalability.....	8
2.3. Reliability.....	9
2.4. Usability.....	9
2.5. Supportability.....	9
2.6. Security.....	9
Data Dictionary.....	10
Use Case Diagrams.....	11
Use Case Descriptions.....	12
UI Mockups.....	42

Introduction/Mission statement

Our mission is to empower amateur cyclists with a seamless and stress-free riding experience by developing a mobile application that enables them to plan safe and scenic cycling routes using park connector networks and view available amenities such as water coolers, bike shops and bike parking spots along the way. They can also connect with fellow cyclists, have favourite routes “saved” and view other cyclists’ activities. This project will be considered completed when the application has been tested and approved by the ZEA. This project supports the goal of the Smart Nation movement to elicit innovative applications that exploit the publicly available government data.

Documentation of Functional and Non-Functional Requirements

1.Functional Requirements

1.1. User Login

- 1.1.1. CyclistGoWhere! should display a Welcome page to allow Users to log into existing accounts.
- 1.1.2. Users should be allowed to log in with their username, or email, followed by the correct password.
 - 1.1.2.1. If an User chooses to log in with their email address, the input text in the “Username/Email” field should be of the correct email format (e.g. user01@gmail.com) and should be less than 512 characters.
 - 1.1.2.2. If an User chooses to log in with their username, the input text in the “Username/Email” should be of at least one character and less than 128 characters.
- 1.1.3. The login page should allow Users to sign in with their Google accounts.
- 1.1.4. The app should have proper error handling for authentication failures.
 - 1.1.4.1. If an User enters the wrong credentials, the page will display an error to notify the user of invalid credentials.
 - 1.1.4.2. If an User enters the wrong credentials for a maximum of 3 tries or the user has forgotten their password, CyclistGoWhere! should redirect the user to the “Forgot password” page.
 - 1.1.4.3. If an User enters his/her credentials correctly, CyclistGoWhere! should notify the user of a successful login and redirect the user to the “Main page”.

1.2. Forgot Password

- 1.2.1. Users should be able to reset their passwords by entering their email.
- 1.2.2. The app should send a temporary password via email to the email address entered in the “Email ID” field.
- 1.2.3. The temporary password should be randomly generated and comply with the password policy stated in 1.3.4.3.

1.3. User Registration

- 1.3.1. CyclistGoWhere! Welcome page should also have an option for Users to create an account.
- 1.3.2. If a New User clicks the “Create Account” button in the Welcome page, CyclistGoWhere! should redirect the user to the Account Registration page.
- 1.3.3. The Account Registration page should allow users to sign up for an account with their Google account.
- 1.3.4. New Users should only be allowed to input the following into the respective fields:
 - 1.3.4.1. Usernames should be of at least one character and less than 128 characters, as well as not containing symbols other than “.” or “_”.
 - 1.3.4.2. Emails should be of valid format (e.g. user01@gmail.com)
 - 1.3.4.3. Passwords should contain 1 uppercase letter, 1 lowercase, at least 1 symbol (!@#\$\$%^&*), and at least 8 characters long.
- 1.3.5. The Account Registration page should have proper error handling for listing 1.2.4.
 - 1.3.5.1. If a New User enters an username that has been used before, CyclistGoWhere! will highlight the username field in “red” and notify the user about the invalid username.
 - 1.3.5.2. If a New User enters an invalid email, CyclistGoWhere! will highlight the email field in “red” and notify the user about the invalid email format.
 - 1.3.5.3. If a New User enters a password that is non-compliant with the password policy stated below the password field, then CyclistGoWhere! should highlight the password field in “red” and notify the user about the invalid password format.
 - 1.3.5.4. If a New User enters a password in the “Confirm your password” that does not match with the password in the “Enter a password” field, then CyclistGoWhere! should notify the user of the password mismatch.

1.4. Navigation Bar

- 1.4.1. The navigation bar must be shown on all interfaces of CyclistGoWhere! at all times, after the user has successfully authenticated.
- 1.4.2. The navigation bar must allow users to navigate to the Main page, Profile page, Settings page.
- 1.4.3. If a user clicks onto the “Home” icon, the app must display the Main Page.
- 1.4.4. If a user clicks onto the “Crew” icon, the app must display the Crew Page.
- 1.4.5. If a user clicks onto the “Profile” icon, the app must display the Profile Page.
- 1.4.6. If a user clicks onto the “Heart” icon, the app must display the Favorites Page.
- 1.4.7. If a user clicks onto the “Gear” icon, the app must display the Settings Page.

1.5. Main Page

1.5.1. Route Search Bar

- 1.5.1.1. Users should be able to view the top 3 most recently searched routes should be displayed in a drop-down box.
- 1.5.1.2. The search function must allow the user to search for specific destinations, landmarks or facilities.
- 1.5.1.3. Upon searching a destination, possible routes will be shown for the user to pick from.
- 1.5.1.4. Routes shown will also include details such as distance and estimated time needed for completion.
- 1.5.1.5. As a user is typing in their destination in the search bar, possible destinations based on autocomplete will be shown below the search bar.
- 1.5.1.6. If a user has inputted an inaccessible destination, a route to an accessible location that is nearest to the destination inputted is shown.

1.5.2. Map

- 1.5.2.1. The main page must be able to display the map of Singapore.
- 1.5.2.2. The map on the main page should be centered and zoomed in on the user’s current location upon loading.
- 1.5.2.3. Park connector routes are highlighted for visibility.
- 1.5.2.4. Map display should include a “Filter” button to select cycling related features such as water coolers, bike repair shops and bike parking stops.
- 1.5.2.5. Selected cycling related facilities using the filter function should be highlighted as their respective icons on the map.

1.5.3. Filters

- 1.5.3.1. The Filter page must allow users to select cycling related facilities such as water coolers, bike parking stops and bike repair shops to only display facilities based on their preferences on the map at any point in time.
- 1.5.3.2. Users can select multiple of their preference data at once, or none.
- 1.5.3.3. If a user selects the “Water Coolers” Filter, the map should show ALL water coolers available.
- 1.5.3.4. If a user selects the “Bike Repair” Filter, the map should show ALL Bike repair shops available.
- 1.5.3.5. If a user selects the “Bike Park” Filter, the map should show ALL Bicycle parks available.
- 1.5.3.6. If a user selects the ALL Filters, the map should show ALL Filters.
- 1.5.3.7. If a user selects none of the Filters, the map should not show any Filters.
- 1.5.3.8. The Filter page must display an alert to the user to confirm any changes to the Filter preferences.

1.5.4. Route Navigation

- 1.5.4.1. The route navigation must have a Start and an End button.
- 1.5.4.2. The route navigation must track the user’s current location.
- 1.5.4.3. The route navigation must reflect accurately on the map the user’s current location.
- 1.5.4.4. Turn-by-turn navigations must be shown throughout the ride.
- 1.5.4.5. Based on the user’s preferences key stops along the way such as water coolers, repair shops and parking spots will be highlighted with their respective icons and no other icons that are not along the way of the route should be displayed.
- 1.5.4.6. The route navigation upon completion must direct users to a pop-up page displaying their activity statistics, including the distance travelled, time elapsed, calories burnt, cadence and elevation gain.
- 1.5.4.7. Route navigation should include a Weather Feature that estimates the weather conditions along the user’s planned route.
- 1.5.4.8. After a completion of the route, the app should notify the user of the successful completion of the route via a pop-up and the pop-up should include an “OK” button and an “Add route to Favorites” button.

1.5.5. Weather Feature

- 1.5.5.1. The app should check and estimate the weather conditions along a user's planned route.
- 1.5.5.2. If the estimated weather prediction has a high chance of rain, the app should notify the user.
- 1.5.5.3. The app should also notify the user of High UV indexes along the user's planned route.

1.6. Crew Page

- 1.6.1. The Crew Page should display a Search bar.
 - 1.6.1.1. Users are able to search other users via their unique usernames.
 - 1.6.1.2. Users should be able to click onto searched results of a friend to view their profile.
 - 1.6.1.3. Users should be able to view the searched user's recent activities after they click into the searched user's profile.
- 1.6.2. The Crew Page should display Most recent activities of Current Friends.
 - 1.6.2.1. Users can view their friend's recent activities, sorted by newest activities shown at the top.
- 1.6.3. The Crew Page should display Current Friend list.
- 1.6.4. Users can add and remove friends by clicking or unclicking a plus button found beside a user's profile.
- 1.6.5. Users can 'thumbs-up' their friends' ride activities.
- 1.6.6. Users can 'heart' (will appear in Saved Page) their friends' ride activities.
- 1.6.7. Users can Try their friend's route by clicking the "Try Route" button.

1.7. Profile Page

- 1.7.1. The profile page must display the user's cycling statistics including total distance travelled within that week, month or year and the total number of rides.
- 1.7.2. Users can see their personal bests for specific distances.
- 1.7.3. Users can view their top 3 most recent activities.
- 1.7.4. Upon clicking on the "See more" button below the recent activities, users can view ALL their past rides history.
- 1.7.5. Users can delete past activities.
- 1.7.6. Users can save past routes.
- 1.7.7. Users can "thumbs-up" their own activities.
- 1.7.8. Users can view the number of "thumbs-up" on their activities.

1.8. Favorites Page

- 1.8.1. The saved routes feature must display ALL saved routes of a user.

- 1.8.2. Users must be able to click onto any of their saved routes to start a cycling activity.
- 1.8.3. Users must be able to delete saved routes directly.

1.9. Settings Page

- 1.9.1. The Settings Page must allow users to either change their Account Settings or the App Settings.
- 1.9.2. The settings page must include a log out button for users to log out of their account.

1.9.3. Account Settings

- 1.9.3.1. CyclistGoWhere! should allow users to change their usernames, email addresses, passwords, and profile pictures.
- 1.9.3.2. Users should only be allowed to enter unique usernames for the username field if they want to change their usernames.
- 1.9.3.3. Users should only be allowed to enter valid email addresses for the email field if they want to change their email.
- 1.9.3.4. Users should only be allowed to enter sufficiently complex passwords in the password field if they want to change their passwords.
- 1.9.3.5. Users should not be allowed to upload an image format other than .jpg, jpeg, or .png with size more than 2MB and 500x500 pixels if they wish to change their profile pictures.

1.9.4. App Settings

- 1.9.4.1. Users must be able to enable or disable notifications and specify the type of notifications they would like to receive.

2.Non-Functional Requirements

2.1. Performance

- 2.1.1. CyclistGoWhere! must provide responsive performance, with less than 2 seconds latency in retrieving and displaying information for 95% of data retrieval options
- 2.1.2. CyclistGoWhere! must be compatible with the iOS and Android operating systems of the last 10 years
- 2.1.3. CyclistGoWhere! must have a startup time of under 3 seconds upon login?
- 2.1.4. CyclistGoWhere! Must have a response time of 1 seconds following each user input.

2.2. Scalability

- 2.2.1. CyclistGoWhere! must handle up to 100 simultaneous users without a degradation in performance.

2.3. Reliability

- 2.3.1. Users do not need to log in again unless they delete the application from their devices or they log out.
- 2.3.2. System must be ready to use at all times unless when there is scheduled maintenance

2.4. Usability

- 2.4.1. Users must successfully locate nearby available facilities (water cooler, repair shops, parking spots).
- 2.4.2. Users should navigate between pages using the application's icons with ease.

2.5. Supportability

- 2.5.1. Database must be replaceable with any commercial product supporting standard SQL / NoSQL

2.6. Security

- 2.6.1. Users must be authenticated when logging in
- 2.6.2. All sensitive information such as passwords and current location should be encrypted

Data Dictionary

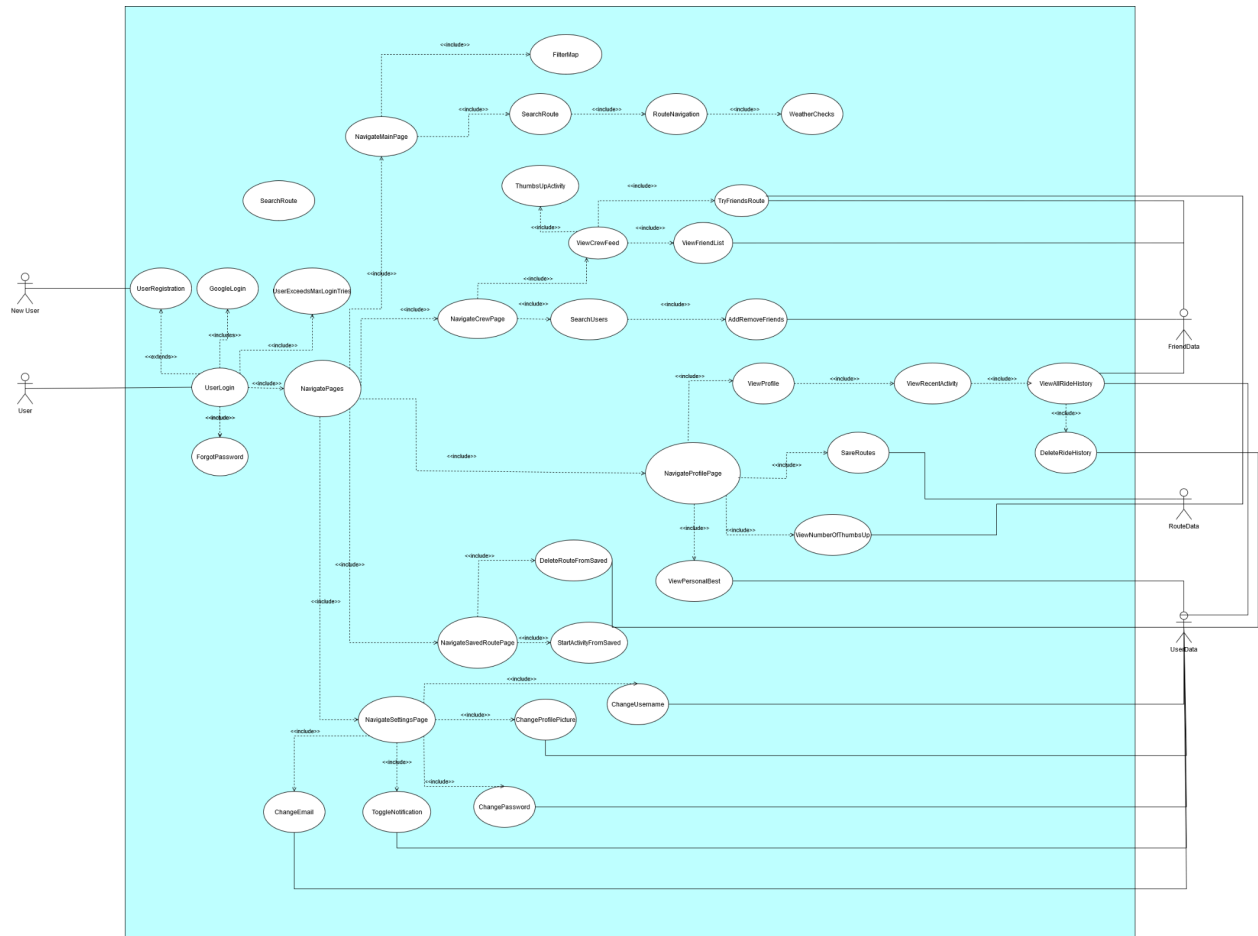
Categories	Terms	Definition	Data Type
Administrative (User)	Username	Users / Admins names when registering for the application	String
	Email	User's email address for registration.	String
	Password	Password for accounts registered	String
	Location	Where the user currently is.	String
Navigation	Destination	Where the user is headed.	String
	Map	An interactive map that displays the user's current geographical location in real time. The map highlights park connector routes and shows available amenities along the user's route.	JSON
	Heart	The symbol of the button that users click to "save" their past routes	Boolean
	Saved Routes	The list of past routes that the user "saved"	JSON
Weather	Weather	Current weather along the route to the destination	JSON
Facilities	Nearby Water Coolers	Location of nearby water coolers along the way	String
	Nearby Repair Shops	Location of nearby repair shops along the way	String
	Repair Shop Name	Name of repair shops	String
	Nearby Parking Spots	Location of nearest parking spots along the way	String
Interaction	Thumbs-Up	User "likes" activity of a friend	Boolean
	Likes	The number of thumbs-ups a user receives on an activity	Integer
Account	Profile	A registered user's personal profile with the	String

		user's past activities and statistics of the user	
	Crew Page	A list of the user's friends and their activities can be seen from this page	-
	Settings	A list of options which the users can personalise and adjust various options and preferences of the app to fit their specific needs and ways of working	-

Use Case Diagrams

Link:

https://app.diagrams.net/#G1NaLrn4uFyc5cfBiJbHn1h1tBbID6_ctY#%7B%22pageId%22%3A%22C5RBs43oDa-KdzZeNtuy%22%7D



Use Case Descriptions

Use Case ID:	1.1.2		
Use Case Name:	UserLogin		
Created By:	Harikrishnan Vinod	Last Updated By:	Harikrishnan Vinod
Date Created:	27/01/2025	Date Last Updated:	27/01/2025

Actor:	Returning User
Description:	This use case will allow the user to log in to their account using Email ID/Username and password
Preconditions:	<ol style="list-style-type: none">1. The user must have the app installed2. The user must have a valid account
Postconditions:	The User will be logged into their account and is navigated to the main page of the application
Priority:	High
Frequency of Use:	1 time upon download and after 3 months of inactivity
Flow of Events:	<ol style="list-style-type: none">1. The User will be prompted at the Welcome Page if they have an existing account or if they are a first time user2. The User upon selection of “Already have an Account” will log in by entering the “Email ID or Username” and “Password” fields3. If the login is successful, the User will be directed to the Main Page of the App
Alternative Flows:	<p>AF-S2: User chooses to login with their Google Account:</p> <ol style="list-style-type: none">1. The User selects the “Login with Google Account” button.2. The user logs into the system using their Google email credentials using the use case Google Account Login <p>AF-S3: User does not remember his password</p> <ol style="list-style-type: none">1. User selects the “Forget Password” button and is directed to the Forget Password use case
Exceptions:	<ol style="list-style-type: none">1. If the login credentials are invalid, an error message is flashed and the user will not be logged in.
Includes:	GoogleLogin ForgetPassword
Special Requirements:	The system needs to validate the user login credentials
Assumptions:	<ol style="list-style-type: none">1. The User has an existing account2. The User is connected to the Internet
Notes and Issues:	None

Use Case ID:	1.1.3
--------------	-------

Use Case Name:	GoogleLogin		
Created By:	Harikrishnan Vinod	Last Updated By:	Harikrishnan Vinod
Date Created:	27/01/2025	Date Last Updated:	27/01/2025

Actor:	First time User or Returning User
Description:	This use case will allow the user to log in to their account using their Google Account
Preconditions:	<ol style="list-style-type: none"> 1. The user must have the app installed 2. The user must have a Google account
Postconditions:	The User will be logged into their account and is navigated to the main page of the application
Priority:	High
Frequency of Use:	1 time upon download and after 3 months of inactivity
Flow of Events:	<ol style="list-style-type: none"> 1. The User will be prompted at the Welcome Page if they have an existing account or if they are a first time user 2. The User upon selection of “Already have an Account” will log in by selecting the “Login using Google Account” button 3. The user is directed to the Google webpage for login 4. If the login is successful, the User will be directed to the Main Page of the App
Alternative Flows:	AF-S4:Google login was not successful <ol style="list-style-type: none"> 1. The system will redirect the User to the App login page and display an error message that the login with Google was unsuccessful
Exceptions:	None
Includes:	None
Special Requirements:	Access to Google Login API
Assumptions:	<ol style="list-style-type: none"> 1. The User has a Google Account
Notes and Issues:	None

Use Case ID:	1.1.4.2		
Use Case Name:	UserExceedsMaxLoginTries		
Created By:	Harikrishnan Vinod	Last Updated By:	Harikrishnan Vinod
Date Created:	27/01/2025	Date Last Updated:	27/01/2025

Actor:	Returning User
Description:	This use case will allow the user to regain access to their account
Preconditions:	<ol style="list-style-type: none"> 1. The use must have a valid account

Postconditions:	The User will be sent a link to their email account that can help them reset their password and regain access to their account
Priority:	High
Frequency of Use:	Almost Never
Flow of Events:	<ol style="list-style-type: none"> 1. The User will be prompted at the Welcome Page if they have an existing account or if they are a first time user 2. The User upon selection of “Already have an Account” will log in by entering the “Email ID or Username” and “Password” fields 3. If the login is unsuccessful after 3 tries, the User will be directed to the Forget Password page 4. The user will be asked to fill in their email address 5. If it is a valid email address for which there is an account, a link will be sent to the User’s email address 6. The user will be able to set a new password and can resume to the User Login use case
Alternative Flows:	None
Exceptions:	<ol style="list-style-type: none"> 1. No valid account associated with the email address is found 2. User is then prompted to go to the Register User use case
Includes:	ForgotPassword
Special Requirements:	The system needs to be able send emails with a reset password link after finding the User’s account
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.2		
Use Case Name:	ForgotPassword		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	28/01/2025	Date Last Updated:	28/01/2025

Actor:	Users
Description:	Users can to reset their passwords by entering their email address
Preconditions:	Users must have an account on the app
Postconditions:	User’s password reset and temporary password sent to their email address
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. User cannot remember their log in password 2. User click onto the “Forgot Password” button in the login page

	<ol style="list-style-type: none"> The app brings the user to the “Forgot Password” page and prompts user to enter his/her email address in the email field User enters his/her email address that was registered in the app The app verifies if the email address entered is an registered email If the email address is a valid registered email address, the app generates a temporary password and set the user’s password as the generated password in the user database The app sends the generated temporary password to the user’s email address
Alternative Flows:	<p>AFS6: If the user enters an email address that is not registered</p> <ol style="list-style-type: none"> The app returns to step 3 The app highlights the Username entry box in red The app displays the message “Invalid username format! Please enter a username not containing any symbols except ‘.’ and ‘_!’” under the entry box
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	The app is able to connect to the database server 100% of the time
Notes and Issues:	None

Use Case ID:	1.3		
Use Case Name:	UserRegistration		
Created By:	Harikrishnan Vinod	Last Updated By:	Harikrishnan Vinod
Date Created:	27/01/2025	Date Last Updated:	27/01/2025

Actor:	First time User
Description:	This use case will allow the user to register for an account
Preconditions:	<ol style="list-style-type: none"> The user must have the app installed The user must have an email account they have access to
Postconditions:	The User will be logged into their account and is navigated to the main page of the application
Priority:	High
Frequency of Use:	1 time
Flow of Events:	<ol style="list-style-type: none"> The User will be prompted at the Welcome Page if they have an existing account or if they are a first time user The User upon selection of “Create an account” will register by entering a valid email account, a username that has not been taken and a password that matches the requirements

	3. If all the requirements are met a valid account is created for the User, he can now login using the User Login use case
Alternative Flows:	AF-S2: User chooses to login with their Google Account: 1. The User selects the “Login with Google Account” button. 2. The user logs into the system using their Google email credentials using the use case Google Account Login
Exceptions:	1. If any required information is invalid or missing, the field will be highlighted in red and an error message is displayed
Includes:	GoogleLogin
Special Requirements:	The system needs to validate the user input data
Assumptions:	1. The User has working email account
Notes and Issues:	None

Use Case ID:	1.4.3		
Use Case Name:	NavigateMainPage		
Created By:	Bryan Goh	Last Updated By:	Cheryl Chin
Date Created:	30 January 2025	Date Last Updated:	2 February 2025

Actor:	User
Description:	The default page is the main page after successful log-in, where a map of Singapore zoomed in on the user’s current location will be displayed. This map highlights park connector routes and the available amenities (water coolers, bike repair shops, bike parking spots). This page provides access to recently saved routes. Users can view their current location and apply filters to display relevant cycling facilities.
Preconditions:	<ol style="list-style-type: none"> 1. User is authenticated and logged in. 2. The device has internet connection and GPS functionality. 3. Location services are enabled, and the app has permission to access location services.
Postconditions:	<ol style="list-style-type: none"> 1. Map is displayed and centered on the user’s current location. 2. Cycling-related amenities (water coolers, bike repair shops and bike parking stops) they have selected are indicated by their respective icons on the map, based on the applied filters. 3. Users can access their recent saved routes.
Priority:	High
Frequency of Use:	High

Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users are navigated to the Main page by default. 3. Map is loaded and zoomed in on the user's current location. 4. Nearby cycling-related amenities (water coolers, bike-repair shops, bike parking spots) will be displayed as their respective icons. 5. Park connector routes on the map are highlighted. 6. Users click on the "Saved" to access their 3 most recently saved cycling routes.
Alternative Flows:	AF-S3: Application is unable to access user's location due to denied permission to access location <ol style="list-style-type: none"> 1. Pop-up to prompt the user to enable location services for CyclistGoWhere!
Exceptions:	EX1: User's location cannot be detected. <ol style="list-style-type: none"> 1. Error message is displayed "Location cannot be detected".
Includes:	FilterMap Map API
Special Requirements:	Phone permission for location access is enabled. The map should be responsive and interactive
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.4.4		
Use Case Name:	NavigateCrewPage		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	30 January 2025	Date Last Updated:	30 January 2025

Actor:	User
Description:	The User can access the Crew Page via clicking onto the "Crew" icon in the Navigation Bar
Preconditions:	User is authenticated and logged in and is in other pages other than the Crew Page
Postconditions:	Crew Page is displayed
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The User is in a page other than the Crew Page. 2. Users click onto the "Crew" icon in the Navigation Bar at the bottom of the screen. 3. The app displays the Crew Page. 4. User scrolls through the Crew page to view friends' activities

Alternative Flows:	None
Exceptions:	None
Includes:	SearchUsers, ViewAllFriendsActivities, ViewFriendList, AddRemoveFriends, ThumbsUpActivity, TryFriendsRoute
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.4.5		
Use Case Name:	NavigateProfilePage		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	30 January 2025	Date Last Updated:	30 January 2025

Actor:	User
Description:	The User can access the Profile Page via clicking onto the “Profile” icon in the Navigation Bar
Preconditions:	User is authenticated and logged in and is in other pages other than the Profile Page
Postconditions:	Profile Page is displayed
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The User is in a page other than the Profile Page. 2. Users click onto the “Profile” icon in the Navigation Bar at the bottom of the screen. 3. The app displays the Profile Page.
Alternative Flows:	AF-3: User has no friends. <ol style="list-style-type: none"> 1. Error message is displayed “No activities displayed.”
Exceptions:	None
Includes:	ViewProfile, ViewPersonalBest, ViewRecentActivity, ViewAllRideHistory, DeleteRideHistory, SaveRideHistory, ThumbsUpRideHistory, ViewNumberOfThumbsUp
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.4.6
Use Case Name:	NavigateSavedRoutesPage

Created By:	Bryan Goh	Last Updated By:	Harikrishnan Vinod
Date Created:	30 January 2025	Date Last Updated:	2 February 2025

Actor:	User
Description:	The User can access the Saved Routes Page via clicking onto the “heart” icon in the Navigation Bar
Preconditions:	User is authenticated and logged in and is in other pages other than the Saved Routes Page
Postconditions:	Saved Routes Page is displayed
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The User is in a page other than the Saved Routes Page. 2. Users click onto the “heart” icon in the Navigation Bar at the bottom of the screen. 3. The app displays the Saved Routes Page. 4. The app lists all saved routes by the User.
Alternative Flows:	None
Exceptions:	If the user has no saved routes, it displays you have “ No routes saved”
Includes:	StartActivityFromFavourites, DeleteRouteFromFavourites
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.4.7		
Use Case Name:	NavigateSettingsPage		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	30 January 2025	Date Last Updated:	30 January 2025

Actor:	User
Description:	The User can access the Settings Page via clicking onto the “Settings” icon in the Navigation Bar
Preconditions:	User is authenticated and logged in and is in other pages other than the Settings Page
Postconditions:	Settings Page is displayed
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The User is in a page other than the Settings Page. 2. Users click onto the “Settings” icon in the Navigation Bar at the bottom of the screen. 3. The app displays the Settings Page.

Alternative Flows:	None
Exceptions:	None
Includes:	ChangeUsername, ChangeEmail, ChangePassword, ChangeProfilePicture, ToggleNotification
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.5.1		
Use Case Name:	SearchRoute		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	30 January 2025	Date Last Updated:	30 January 2025

Actor:	User
Description:	In the Main Page, the User should be able to search for their “FROM” and “TO” addresses, then the app will find a cycling path for them when they click on the “GO” button.
Preconditions:	User is authenticated and logged in and in the Main Page
Postconditions:	User successfully search for their cycling route
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account and is navigated to the Main Page. 2. The app will display “FROM” and “TO” search bars followed by a map that shows their current location. 3. The app will automatically populate the “FROM” search bar with their detected current location. 4. User clicks onto the “FROM” search bar. 5. The app displays their top 3 most recently searched routes in the drop-down list that users can choose to click for quicker search. 6. If the User searches their desired location without clicking onto any recent past routes, the app will prompt the user to enter the “TO” location by shifting the focus onto the “TO” search bar. 7. The User enters their destination in the “TO” search bar and clicks onto the “GO” button. 8. The app then searches for a cycling route based on the “TO” and “FROM” locations and displays it on the map.
Alternative Flows:	AF-S6: User clicks onto any of their past history in the drop-down list

	<ol style="list-style-type: none"> 1. The app automatically populates the “FROM” and “TO” fields and searches for a route. 2. The app then displays the route on the map.
Exceptions:	EX1: User’s location cannot be found <ol style="list-style-type: none"> 1. Error message is displayed “Could not find location!”
Includes:	Map API, RouteNavigation
Special Requirements:	Phone permission for location access is enabled
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.5.3		
Use Case Name:	FilterMap		
Created By:	Bryan Goh	Last Updated By:	Cheryl Chin
Date Created:	30 January 2025	Date Last Updated:	2 February 2025

Actor:	User
Description:	Users can select cycling related facilities such as water coolers, bike parking stops and bike repair shops to only display facilities based on their preferences on the map at any point in time.
Preconditions:	User is authenticated and logged in and navigated to the Main Page
Postconditions:	User successfully filtered for preferred facilities on the map
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account and is navigated to the Main Page. 2. The app will display the map with no filters on the Main Page by default. 3. User clicks onto the “Filter” button on the bottom left corner of the map. 4. The app will display “Water Coolers”, “Bike Repair”, and “Bike Park” checkboxes. 5. If the user checks the “Water Coolers” checkbox and no other checkboxes, then clicks “Confirm”, the map will display ALL water coolers on the map.
Alternative Flows:	AF-S5: User checks ALL filters available <ol style="list-style-type: none"> 1. User checks on the “Water Coolers”, “Bike Repair”, and “Bike Park” checkboxes. 2. The map will display ALL water coolers, bike repair shops, and bicycle parks on the map.

Exceptions:	EX1: If no cycling-related facilities match the selected filters, the system displays a message: “No results found.”
Includes:	Map API
Special Requirements:	Phone permission for location access is enabled
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.5.4		
Use Case Name:	RouteNavigation		
Created By:	Cheryl Chin	Last Updated By:	Cheryl Chin
Date Created:	2/2/25	Date Last Updated:	2/2/25

Actor:	User
Description:	Users are able to press the start button to begin navigating a route, receive real-time turn-by-turn directions, and track key stops such as water coolers, bicycle repair shops and parking spots along their route
Preconditions:	Users must be authenticated and logged in. Location services must be enabled. The device must have GPS functionality with internet connection active. Users are navigated to the main page.
Postconditions:	Upon completing the route, the user can see a summary of their activity, which denotes a map of the route they travelled, along with statistics such as their distance travelled, time elapsed, calories burnt, cadence and elevation gain, below the map.
Priority:	High
Frequency of Use:	Multiple times a day (depend on user’s activity level and frequency of use)
Flow of Events:	<ol style="list-style-type: none"> 1. User presses the “Start” button. 2. The system initialises GPS and it starts to track the user’s location. 3. The map displays the user’s current location and navigation route. 4. A weather feature that estimates weather conditions along the user's planned route is shown. 5. Turn-by-turn navigation instructions will be provided through a bar at the top section of the screen.

	<ol style="list-style-type: none"> Key stops (water coolers, bicycle repair shops and parking spots) will be indicated on the map with their respective icons). No other icons that are not along the way of the route will be displayed. User presses the “End” button upon reaching their destination. The app will notify users of successful route completion via a pop-up and the pop-up should include an “OK” and “Saved Route” buttons. The system will bring users to another pop-up page with all their activity statistics.
Alternative Flows:	AF-S1: If GPS functionality is not available where they are, the system will show an offline error, and prompt the user to enable location services.
Exceptions:	None
Includes:	Key stops (water coolers, bicycle repair shops and parking spots) will be indicated on the map with their respective icons), Map API, WeatherChecks
Special Requirements:	The interface must be user friendly and accessible GPS tracking needs to be accurate and the map needs to be timely updated with new/closed spots.
Assumptions:	Users will only navigate in areas with Internet connectivity and GPS connectivity.
Notes and Issues:	None

Use Case ID:	1.5.5		
Use Case Name:	WeatherChecks		
Created By:	Ethan	Last Updated By:	Ethan
Date Created:	28/1/25	Date Last Updated:	28/1/25

Actor:	User
Description:	Application will display alerts for users when there is inclement weather such as rain or high uv index detected within 3 hours for the desired route to be taken.
Preconditions:	Users have already used RouteNavigation to obtain a path to their destination.
Postconditions:	None
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> User sets a route by the RoutingFunction. If there is a chance of rain or high uv index along the routed path, the application will notify the user of the bad weather and ask if they are willing to proceed with the current path.

Alternative Flows:	None
Exceptions:	None
Includes:	SearchRoute, RouteNavigation
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.6.1.1 - 1.6.1.2		
Use Case Name:	SearchUsers		
Created By:	Lim En Jia	Last Updated By:	Lim En Jia
Date Created:	31 January 2025	Date Last Updated:	31 January 2025

Actor:	User
Description:	To search other cyclists' profiles.
Preconditions:	<ol style="list-style-type: none"> 1. User is authenticated and logged in. 2. User navigates to the Crew Page using the navigation toolbar at the bottom of the pages.
Postconditions:	Users find other cyclists' profiles.
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users navigate to the Crew page using the navigation toolbar at the bottom of the page. 3. Users key in usernames of other users in the user search bar to search for their profiles. 4. Users can click onto searched results to view other users' profiles.
Alternative Flows:	AF-3: Username does not exist. <ol style="list-style-type: none"> 1. Error message is displayed "User does not exist."
Exceptions:	None
Includes:	List of all users
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.6.3		
Use Case Name:	ViewFriendList		
Created By:	Lim En Jia	Last Updated By:	Lim En Jia
Date Created:	31 January 2025	Date Last Updated:	31 January 2025

Actor:	User
Description:	To view a list of all of User's friends.
Preconditions:	<ol style="list-style-type: none"> 1. User is authenticated and logged in. 2. User navigates to the Crew Page using the navigation toolbar at the bottom of the pages.
Postconditions:	User views all of his/her friends.
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users navigate to the Crew page using the navigation toolbar at the bottom of the page. 3. Users scroll through the friend list to view all friends.
Alternative Flows:	AF-3: User has no friends. <ol style="list-style-type: none"> 1. Error message is displayed "No friends."
Exceptions:	None
Includes:	List of all friends (including their usernames and profile pictures)
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.6.4		
Use Case Name:	AddRemoveFriends		
Created By:	Lim En Jia	Last Updated By:	Lim En Jia
Date Created:	28 January 2025	Date Last Updated:	31 January 2025

Actor:	User
Description:	To add or remove friends.
Preconditions:	<ol style="list-style-type: none"> 1. User is authenticated and logged in. 2. User navigates to the Crew Page using the navigation toolbar at the bottom of the pages.
Postconditions:	<ol style="list-style-type: none"> 1. Users successfully add or remove friends. 2. Users view friends' ride activities. 3. "Thumbs-up" for ride activities saved and reflected on the page.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users navigate to the Crew page using the navigation toolbar at the bottom of the page.

	<ol style="list-style-type: none"> Users key in usernames of other users in the search bar to search for their profiles. Users can click on the plus button beside users' profiles to add them as friends.
Alternative Flows:	<p>AF-S3: Username not found</p> <ol style="list-style-type: none"> Empty page with message displayed "Username does not exist!" <p>AF-S4: Remove friends</p> <ol style="list-style-type: none"> Users can click the plus button again to unfriend users. Plus button will change colour from Green (friends) to Grey (unfriend).
Exceptions:	None
Includes:	List of all users, SearchUsers, ViewProfile
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.6.5		
Use Case Name:	ThumbsUpActivity		
Created By:	Lim En Jia	Last Updated By:	Lim En Jia
Date Created:	31 January 2025	Date Last Updated:	31 January 2025

Actor:	User
Description:	To "thumbs-up" activities.
Preconditions:	<ol style="list-style-type: none"> User is authenticated and logged in. User navigates to the Crew Page or Profile Page using the navigation toolbar at the bottom of the pages.
Postconditions:	<ol style="list-style-type: none"> Users "thumbs-up" activities.
Priority:	Low
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> The User successfully logs into his/her account. Users navigate to the Crew page using the navigation toolbar at the bottom of the page. Users can view their friends' activities, sorted by newest activities first. Users click the "thumbs-up" button to like their friends' activities.
Alternative Flows:	<p>AF-S2: Users navigate to Profile Page instead.</p> <ol style="list-style-type: none"> Users click the "See More" button to view all ride histories.

	2. Users click the “thumbs-up” button to like their own activities. AF-S3: User search for other users’ profiles instead. <ol style="list-style-type: none"> 1. Users key in usernames of other users in the user search bar to search for their profiles. 2. Users click into other cyclists’ profiles. 3. Users click the ‘thumbs-up’ button to like their friends’ activities. AF-S4: Remove “thumbs-up” activities <ol style="list-style-type: none"> 1. Users can click the “thumbs-up” button again to unlike activities. 2. “Thumbs-up” button will change colour from Green (liked) to Grey (not liked).
Exceptions:	EX1: No friends added initially so the Crew page is unable to display any friends’ activities. <ol style="list-style-type: none"> 1. Empty page with message displayed “Get connected with other users now!”
Includes:	SearchUsers, ViewCrewFeed, ThumbsUpRideHistory, ViewProfile
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.6.6 / 1.7.6		
Use Case Name:	SaveRoutes		
Created By:	Lim En Jia	Last Updated By:	Lim En Jia
Date Created:	31 January 2025	Date Last Updated:	31 January 2025

Actor:	User
Description:	To save routes.
Preconditions:	<ol style="list-style-type: none"> 1. User is authenticated and logged in. 2. User navigates to the Crew Page or Profile Page using the navigation toolbar at the bottom of the pages.
Postconditions:	Users saved friends' activities.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users navigate to the Crew page using the navigation toolbar at the bottom of the page. 3. Users can scroll through their friends' activities, sorted by newest activities first. 4. Users click the “heart” buttons to save routes.
Alternative Flows:	AF-S2: Navigate to Profile Page <ol style="list-style-type: none"> 1. Scroll down to view top 3 recent activities.

	<ol style="list-style-type: none"> Click the “See More” button to view all ride activities. Users click the “heart” buttons to save routes. <p>AF-S4: Unsave activities</p> <ol style="list-style-type: none"> Users can click the “heart” button again to unsave activities. “Heart” button will change colour from Green (saved) to Grey (not saved). <p>AF-S3: User search for other users’ profiles instead.</p> <ol style="list-style-type: none"> Users key in usernames of other users in the user search bar to search for their profiles. Users click into other cyclists’ profiles. Users click the “heart” button found under their friends’ activities.
Exceptions:	<p>EX1: No friends added initially so the Crew page is unable to display any friends’ activities.</p> <ol style="list-style-type: none"> Empty page with message displayed “Get connected with other users now!”
Includes:	Friends’ Activities, SearchUsers, ViewProfile, ViewCrewFeed
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.6.7		
Use Case Name:	TryFriendsRoute		
Created By:	Lim En Jia	Last Updated By:	Lim En Jia
Date Created:	31 January 2025	Date Last Updated:	31 January 2025

Actor:	User
Description:	To try routes that have been done by friends.
Preconditions:	<ol style="list-style-type: none"> User is authenticated and logged in. User navigates to the Crew Page using the navigation toolbar at the bottom of the pages.
Postconditions:	Users tried friends' routes.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> The User successfully logs into his/her account. Users navigate to the Crew page using the navigation toolbar at the bottom of the page. Users can scroll through their friends' activities, sorted by newest activities first. Users try their friends’ routes.
Alternative Flows:	<p>AF-S4: Users un-try friends’ route.</p> <ol style="list-style-type: none"> Click the ‘cancel’ button found at the top of the screen.

Exceptions:	None
Includes:	SearchUsers, ViewOtherUserActivity, ViewCrewFeed
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.7.1		
Use Case Name:	ViewProfile		
Created By:	Lim En Jia	Last Updated By:	Lim En Jia
Date Created:	28 January 2025	Date Last Updated:	28 January 2025

Actor:	User
Description:	Users can view the total ride distance, total ride time and total number of rides within every week, month and year. Users can view individual activities that made up the period.
Preconditions:	1. User is authenticated and logged in.
Postconditions:	Users view his/her cycling statistics and ride activities for the specified period of time.
Priority:	Low
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users navigate to the Profile Page using the navigation toolbar at the bottom of the page. 3. Cycling statistics and all ride activities for the week (default) are displayed.
Alternative Flows:	AF-S2: View profile of other users. <ol style="list-style-type: none"> 1. Navigate to the Crew Page. 2. Search for other users using the user search bar. 3. Click into other users' profiles to view their statistics.
Exceptions:	EX1: No statistics available <ol style="list-style-type: none"> 1. Empty page with message displayed "Start your first ride today!"
Includes:	SearchUsers, Ride statistics of every users
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.7.2
Use Case Name:	ViewPersonalBest

Created By:	Lim En Jia	Last Updated By:	Lim En Jia
Date Created:	28 January 2025	Date Last Updated:	28 January 2025

Actor:	User
Description:	Users can view their ride achievements for specific distance or speed.
Preconditions:	<ol style="list-style-type: none"> 1. User is authenticated and logged in. 2. User navigates to the Profile Page using the navigation toolbar at the bottom of the pages.
Postconditions:	Users view his/her personal best rides.
Priority:	Low
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users navigate to the Profile Page using the navigation toolbar at the bottom of the page. 3. User scrolls down to the Personal Best section of the Profile Page. 4. Personal best ride activities are displayed according to distance and speed.
Alternative Flows:	None
Exceptions:	EX1: No statistics available <ol style="list-style-type: none"> 1. Empty page with message displayed “Start your first ride today!”
Includes:	Users’ own cycling statistics.
Special Requirements:	Application to compare all activities from users to determine personal bests.
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.7.3		
Use Case Name:	ViewRecentActivity		
Created By:	Bryan Goh	Last Updated By:	Lim En Jia
Date Created:	28 January 2025	Date Last Updated:	28 January 2025

Actor:	Registered User
Description:	Users can view their top 3 most recent activities.
Preconditions:	<ol style="list-style-type: none"> 1. User is authenticated and logged in. 2. User navigates to the Profile Page using the navigation toolbar at the bottom of the pages.
Postconditions:	<ol style="list-style-type: none"> 1. Users are able to view the top 3 most recent past ride activities.

Priority:	Low
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users navigate to the Profile Page using the navigation toolbar at the bottom of the page. 3. Past activities are displayed, with newest activities arranged at the top.
Alternative Flows:	None
Exceptions:	EX1: No statistics available <ol style="list-style-type: none"> 1. Empty page with message displayed “Start your first ride today!”
Includes:	ViewProfile
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.7.4		
Use Case Name:	ViewAllRideHistory		
Created By:	Lim En Jia	Last Updated By:	Lim En Jia
Date Created:	28 January 2025	Date Last Updated:	28 January 2025

Actor:	User
Description:	Users can view all past rides. Users can view the number of “thumbs-up” for ride activities.
Preconditions:	<ol style="list-style-type: none"> 1. User is authenticated and logged in. 2. User navigates to the Profile Page or Crew Page using the navigation toolbar at the bottom of the pages.
Postconditions:	<ol style="list-style-type: none"> 1. Users view past ride activities of self and others. 2. Users view the number of “thumbs-up” for each activity.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users navigate to the Profile Page using the navigation toolbar at the bottom of the page. 3. Past activities are displayed, with most recent 3 activities arranged at the top of the Profile Page. 4. Click the “See More” button below the recent 3 activities to view all past ride history.
Alternative Flows:	AF-S4: No more activities (only 3 activities that are already displayed). <ol style="list-style-type: none"> 1. “See More” button disappears.
Exceptions:	EX1: No statistics available

	1. Empty page with message displayed “Start your first ride today!”
Includes:	SearchUsers, Ride statistics of every user
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.7.5		
Use Case Name:	DeleteRideHistory		
Created By:	Lim En Jia	Last Updated By:	Lim En Jia
Date Created:	28 January 2025	Date Last Updated:	28 January 2025

Actor:	User
Description:	Users can view or delete past rides.
Preconditions:	<ol style="list-style-type: none"> 1. User is authenticated and logged in. 2. User navigates to the Profile Page using the navigation toolbar at the bottom of the pages.
Postconditions:	Users delete certain ride activities.
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users navigate to the Profile Page using the navigation toolbar at the bottom of the page. 3. Past activities are displayed, with newest activities arranged at the top. 4. Users click the “See More” button to view all ride history. 5. Users click the ‘Trash Bin’ icon beside activities to delete them. 6. Database updated to remove activities from user’s profile. 7. The ride history list is updated to remove the deleted activity.
Alternative Flows:	None
Exceptions:	EX1: No statistics available <ol style="list-style-type: none"> 1. Empty page with message displayed “Start your first ride today!”
Includes:	ViewProfile, ViewRecentActivity, ViewAllRideHistory, ViewPersonalBest
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.7.6		
Use Case Name:	SaveRideHistory		
Created By:	Lim En Jia	Last Updated By:	Lim En Jia
Date Created:	28 January 2025	Date Last Updated:	28 January 2025

Actor:	User
Description:	Users can view or delete past rides.
Preconditions:	<ol style="list-style-type: none"> 1. User is authenticated and logged in. 2. User navigates to the Profile Page using the navigation toolbar at the bottom of the pages.
Postconditions:	<ol style="list-style-type: none"> 1. Users save past ride activities.
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users navigate to the Profile Page using the navigation toolbar at the bottom of the page. 3. Past activities are displayed, with newest activities arranged at the top. 4. Users click the “See More” button to view all ride history. 5. Users click the ‘heart’ button beside a past activity to save routes. 6. Database updated to add selected route to Favorites Route. 7. The app adds the saved route to the user’s Favorites Page.
Alternative Flows:	AF-S3: Delete Activities <ol style="list-style-type: none"> 1. Users click the ‘Trash Bin’ button beside activities to delete them. 2. Database updated to remove activities from user’s profile.
Exceptions:	EX1: No statistics available <ol style="list-style-type: none"> 2. Empty page with message displayed “Start your first ride today!”
Includes:	ViewProfile, ViewRecentActivity, ViewAllRideHistory, ViewPersonalBest
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.7.8		
Use Case Name:	ViewNumberOfThumbsUp		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh

Date Created:	28 January 2025	Date Last Updated:	28 January 2025
---------------	-----------------	--------------------	-----------------

Actor:	User
Description:	Users can view or delete past rides.
Preconditions:	<ol style="list-style-type: none"> 1. User is authenticated and logged in. 2. User navigates to the Profile Page using the navigation toolbar at the bottom of the pages.
Postconditions:	Users can view the number of “thumbs-up” for their activities.
Priority:	Low
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. Users navigate to the Profile Page using the navigation toolbar at the bottom of the page. 3. Past activities are displayed, with newest activities arranged at the top. 4. Users click the “See More” button to view all ride history. 5. The number of “thumbs-up” for each activity is displayed beside the “thumbs-up” button.
Alternative Flows:	None
Exceptions:	None
Includes:	ViewProfile ViewRecentActivity ViewAllRideHistory
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

Use Case ID:	1.8.2		
Use Case Name:	StartActivityFromSaved		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	27/01/2025	Date Last Updated:	27/01/2025

Actor:	User
Description:	Allows an User to start a cycling activity directly from their saved routes by clicking onto the saved route displayed on the Favorites page.
Preconditions:	User is authenticated and logged in
Postconditions:	User is able to start a saved route
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account.

	<ol style="list-style-type: none"> The app displays the Main page of the app along with a Navigation Bar on the bottom of the screen. The User clicks on the “heart” button on the navigation bar. The app lists all saved routes by the User. The User clicks onto any of the saved routes. The app routes and starts the cycling activity based on the saved “FROM” and “TO” destinations.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	The app is able to connect to the database server 100% of the time.
Notes and Issues:	None

Use Case ID:	1.8.3		
Use Case Name:	DeleteRouteFromSaved		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	27/01/2025	Date Last Updated:	27/01/2025

Actor:	User
Description:	Allows an User to start a cycling activity directly from their saved routes by clicking onto the saved route displayed on the Favorites page.
Preconditions:	User is authenticated and logged in
Postconditions:	User is able to start a saved route
Priority:	High
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> The User successfully logs into his/her account. The app displays the Main page of the app along with a Navigation Bar on the bottom of the screen. The User clicks on the “heart” button on the navigation bar. The app lists all saved routes by the User. The User clicks onto the “DELETE” button of any of the saved routes. The app deletes the saved route from the user database.
Alternative Flows:	None
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	The app is able to connect to the database server 100% of the time.

Notes and Issues:	None
-------------------	------

Use Case ID:	1.9.3.2		
Use Case Name:	ChangeUsername		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	27/01/2025	Date Last Updated:	27/01/2025

Actor:	User
Description:	Allows an User to change his/her unique user identifier (Username)
Preconditions:	User is authenticated and logged in and navigated to the Settings Page.
Postconditions:	Username of User is changed
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 7. The User successfully logs into his/her account. 8. The app displays the Main page of the app along with a Navigation Bar on the bottom of the screen. 9. The User selects “Account Settings” on the Settings Page. 10. The app prompts the User to select their desired account settings to change: Change Username, Change Email Address, Change Password, Change Profile Picture. 11. The User selects the Change Username option. 12. The app then prompts the User to enter a new username. 13. The User enters an username. 14. The app verifies that the username entered is of valid format. 15. If the user enters a username of valid format, the app queries the database to verify that the username is unique. 16. If the username is valid and unique, the app modifies the username of the User. 17. The app displays a successful change of username and returns to the Account Settings page.
Alternative Flows:	<p>AF-S9: If user enters a username that is not of valid format</p> <ol style="list-style-type: none"> 1. The app returns to step 6. 2. The app highlights the Username entry box in red. 3. The app displays the message “Invalid username format! Please enter a username not containing any symbols except ‘.’ and ‘_’!” under the entry box. <p>AF-S10: If user enters a non-unique username</p> <ol style="list-style-type: none"> 1. The app returns to step 6. 2. The app highlights the Username entry box in red.

	3. The app displays the message “Username in use. Please enter another unique username!” under the entry box.
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	The app is able to connect to the database server 100% of the time.
Notes and Issues:	None

Use Case ID:	1.9.3.3		
Use Case Name:	ChangeEmail		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	27/01/2025	Date Last Updated:	27/01/2025

Actor:	User
Description:	Allows an User to change his/her Email address
Preconditions:	User is authenticated and logged in and navigated to the Settings Page
Postconditions:	Email address of User is changed
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. The app displays the Main page of the app along with a Navigation Bar on the bottom of the screen. 3. The User selects “Account Settings” on the Settings Page. 4. The app prompts the User to select their desired account settings to change: Change Username, Change Email Address, Change Password, Change Profile Picture. 5. The User selects the Change Email Address option. 6. The app then prompts the User to enter a new email address. 7. The User enters an Email address. 8. The app verifies that the Email address entered is of valid format. 9. If the Email address entered is of valid format, the app modifies the Email address of the User in the database. 10. The app displays a successful change of Email address and returns to the Account Settings page.
Alternative Flows:	AF-S9: If user enters a username that is not of valid format <ol style="list-style-type: none"> 1. The app returns to step 6. 2. The app highlights the Email entry box in red.

	3. The app displays the message “Invalid username format! Please enter a valid email. For example: user01@gmail.com”.
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	The app is able to connect to the database server 100% of the time.
Notes and Issues:	None

Use Case ID:	1.9.3.4		
Use Case Name:	ChangePassword		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	27/01/2025	Date Last Updated:	27/01/2025

Actor:	User
Description:	Allows an User to change his/her login password.
Preconditions:	User is authenticated and logged in and Navigated to the Settings Page
Postconditions:	Password of User is changed
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The User successfully logs into his/her account. 2. The app displays the Main page of the app along with a Navigation Bar on the bottom of the screen. 3. The User selects “Account Settings” on the Settings Page. 4. The app prompts the User to select their desired account settings to change: Change Username, Change Email Address, Change Password, Change Profile Picture. 5. The User selects the Change Password option. 6. The app then prompts the User to enter a new password. 7. The User enters a new password. 8. The app verifies that the password entered is compliant with the password policy. 9. If the user enters a password that is compliant with the password policy, the app updates the password for the user in the database server. 10. The app displays a successful change of username and returns to the Account Settings page.
Alternative Flows:	AF-S9: If user enters a password that is non-compliant with the password policy <ol style="list-style-type: none"> 1. The app returns to step 6.

	<ol style="list-style-type: none"> The app highlights the Password entry box in red. The app displays the message “Password entered is not strong enough! Please enter a password that contains at least 1 uppercase letter, 1 lowercase, 1 symbol (!@#%\$%^&*), and at least 8 characters long!” under the entry box.
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	The app is able to connect to the database server 100% of the time.
Notes and Issues:	None

Use Case ID:	1.9.3.5		
Use Case Name:	ChangeProfilePicture		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	27/01/2025	Date Last Updated:	27/01/2025

Actor:	User
Description:	Allows an User to change his/her profile/display picture.
Preconditions:	User is authenticated and logged in and Navigated to the Settings Page
Postconditions:	Profile picture of User is changed
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> The User successfully logs into his/her account. The app displays the Main page of the app along with a Navigation Bar on the bottom of the screen. The User selects “Account Settings” on the Settings Page. The app prompts the User to select their desired account settings to change: Change Username, Change Email Address, Change Password, Change Profile Picture. The User selects the Change Password option. The app then prompts the User to upload a picture. The User browses through his/her file explorer to select a picture to upload. The app verifies if the file uploaded is of valid file format. If the user uploads a file with the correct file format, the app verifies if the file is of reasonable file size. If the file is of reasonable file size, the app uploads the file into the file server and updates the file path of the Profile picture in the User database.


	11. The app displays a successful change of Profile picture and returns to the Account Settings page.
Alternative Flows:	<p>AF-S9: If user uploads a file of invalid file format</p> <ol style="list-style-type: none"> 1. The app returns to step 6. 2. The app highlights the File Upload entry box in red. 3. The app displays the message “Incorrect file type! Please only upload .jpg, .jpeg, or .png files!” under the entry box. <p>AF-S10: If user uploads a file of invalid file format</p> <ol style="list-style-type: none"> 1. The app returns to step 6. 2. The app highlights the File Upload entry box in red. 3. The app displays the message “File too large! Please upload a file under 2MB!” under the entry box.
Exceptions:	<p>EX1: If the app is not able to locate the file in the file path that the User specified</p> <ol style="list-style-type: none"> 1. The app returns to step 6. 2. The app highlights the File Upload entry box in red. 3. The app displays the message “Could not find <file_path>!”
Includes:	None
Special Requirements:	None
Assumptions:	The app is able to connect to the file server and the database server 100% of the time.
Notes and Issues:	None

Use Case ID:	1.9.4.1		
Use Case Name:	ToggleNotification		
Created By:	Bryan Goh	Last Updated By:	Bryan Goh
Date Created:	27/01/2025	Date Last Updated:	27/01/2025

Actor:	User
Description:	Allows an User to change his/her preference to allow the app to push notifications
Preconditions:	User is authenticated and logged in and navigated to the Settings Page
Postconditions:	Notification preference is toggled
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The User clicks on “App Settings” in the Settings Page. 2. The app displays a Push Notification toggle switch. 3. The User clicks onto the toggle switch

	4. If the Push Notification toggle switch is switched ON initially, the app will turn OFF notification pushing for the user.
Alternative Flows:	AF-S3: If the Push Notification toggle switch is switched OFF initially 1. The app will turn ON notification pushing for the user.
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

UI Mockups

Page Name	UI Mockups
Loading Page	

Login

5:13 PM

Login

Email ID or Username

Password

Forgot Password ?

Login

or with

Don't have an account? [Sign Up](#)

Sign Up

5:13 PM

Sign Up

Username

Email ID

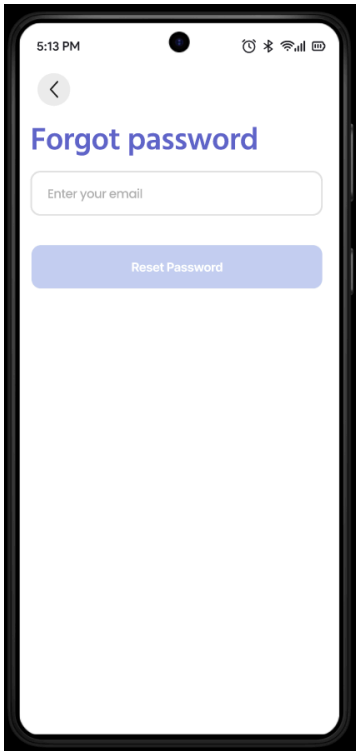
Password

Create

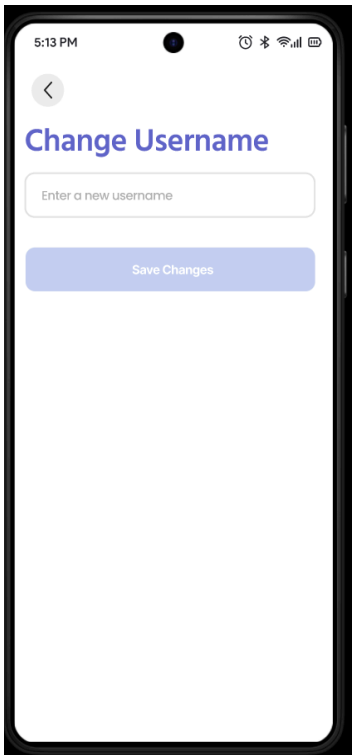
or with

Have any account? [Sign In](#)

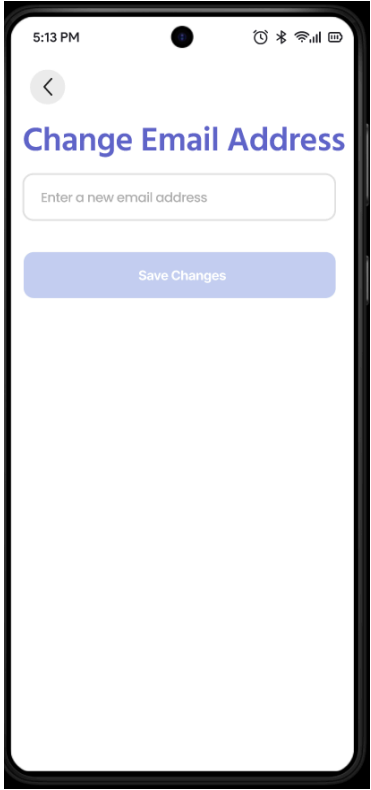
Forget Password



Change Username

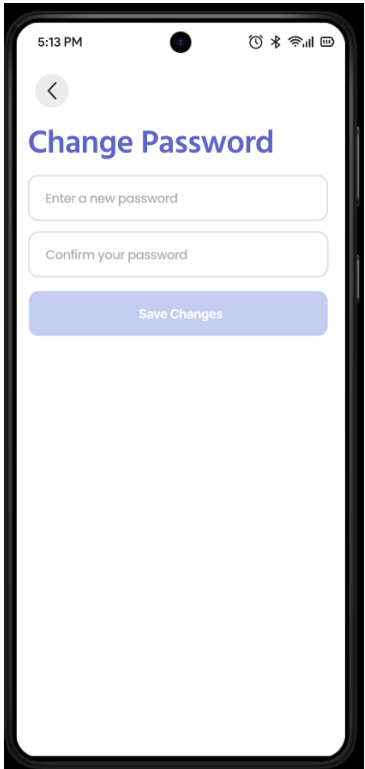


Change Email
Address



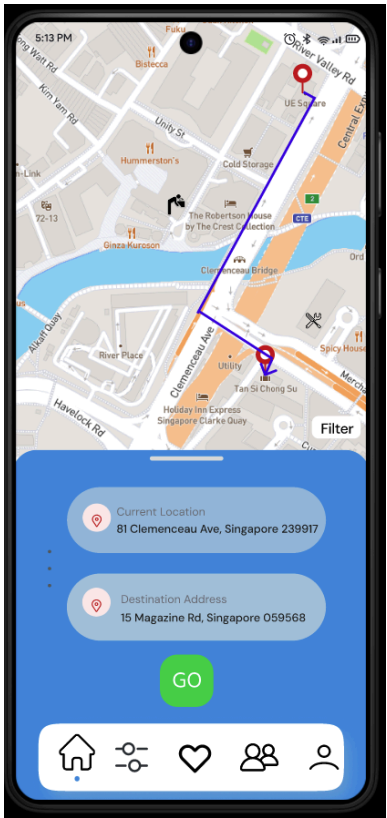
A mobile app screenshot of the 'Change Email Address' screen. The status bar at the top shows the time as 5:13 PM and various system icons. A back arrow is in the top left. The title 'Change Email Address' is in blue. Below it is a text input field with the placeholder 'Enter a new email address'. At the bottom is a blue 'Save Changes' button.

Change Password

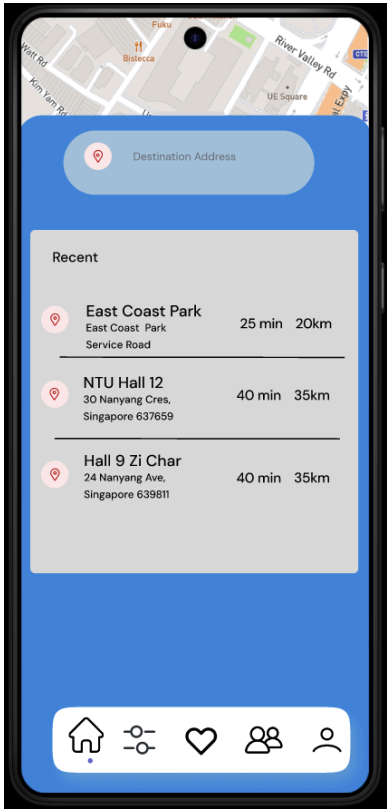


A mobile app screenshot of the 'Change Password' screen. The status bar at the top shows the time as 5:13 PM and various system icons. A back arrow is in the top left. The title 'Change Password' is in blue. Below it are two text input fields: the first with the placeholder 'Enter a new password' and the second with 'Confirm your password'. At the bottom is a blue 'Save Changes' button.

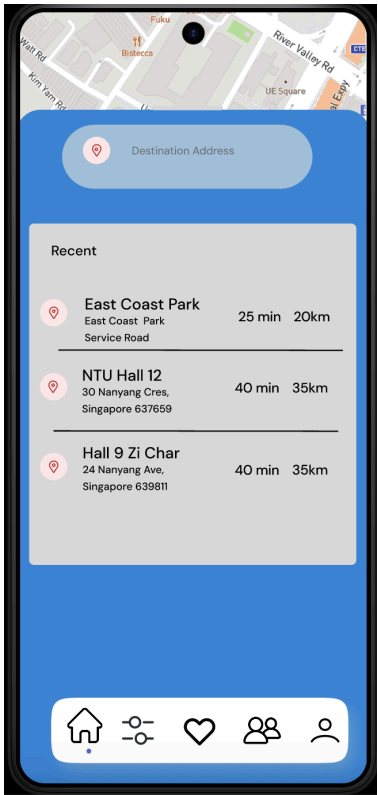
Main



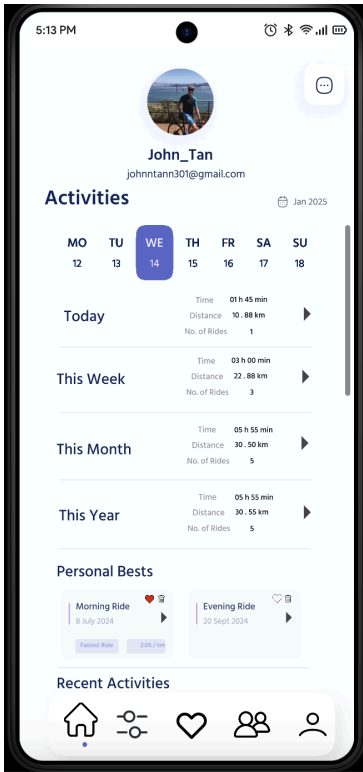
Recent Locations



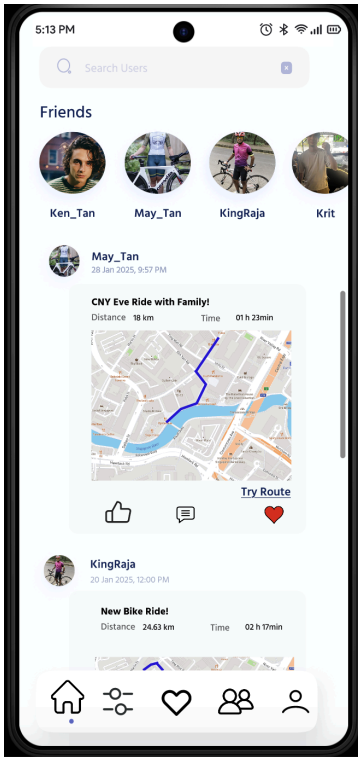
Set Location



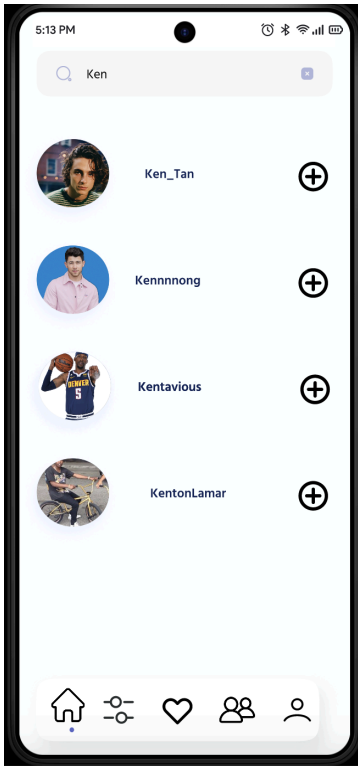
Profile



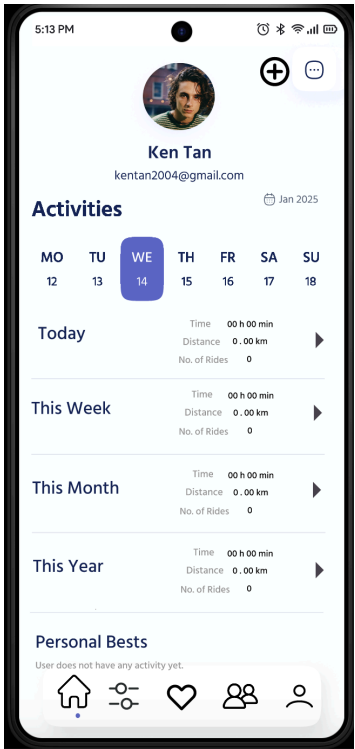
Crew



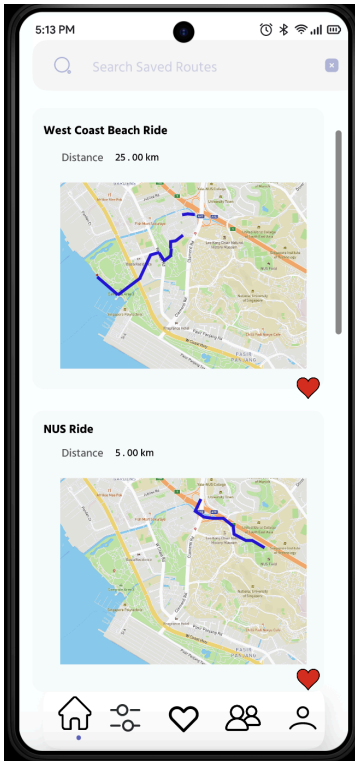
Search User



After Clicking into
Other Users' Profiles



Saved Routes



Settings

