# Software Requirements Specification

**for**

# <Urban>

**Version 1.0 approved**

**Prepared by <Arun Ezekiel, Dinesh, Kai, Ryan, Wei Xian>**

**<SC2001 Software Engineering Z59_Group1>**

**<15/02/2023>**

# Table of Contents

# Revision History

| Name | Date | Reason For Changes | Version |
|---|---|---|---|
| | | | |
| | | | |

# 1. Introduction

## 1.1 Purpose

This document details the software requirements of the android application “Urban”. It defines the feature requirements, use cases and control flow of the application.

## 1.2 Product Scope

“Urban” is a mobile application with an interactive map that tracks a user’s cycling or running sessions when the user embarks on a session, and provides users a community forum to post their sessions. Additionally, the application allows users to view session history, edit goals and edit personal profiles. Users can also comment and like posts on the community platform.

# 2. Overall Description

## 2.1 Product Perspective

“Urban” integrates Google Maps API to provide up-to-date user position on the map. The application also leverages Google Firebase, by storing and displaying user data such as past sessions, goals and profiles. This serves to create a more personalised and intuitive user experience.

## 2.2 Product Functions

Major Functions:

- Login
- Register for a Urban account
- Track session
- View session history
- View, post, comment and like on community forum
- Display map

Minor Functions:

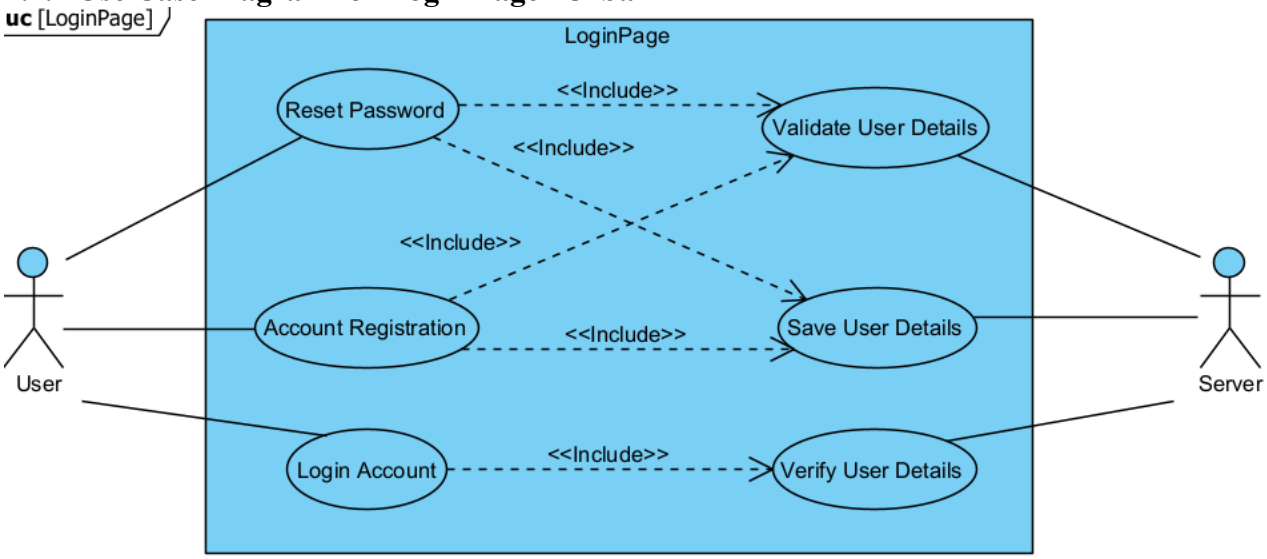
- Reset forgotten password
- Edit goal
- Manage profile
- Pause/Resume live session

## 2.3 Assumptions and Dependencies

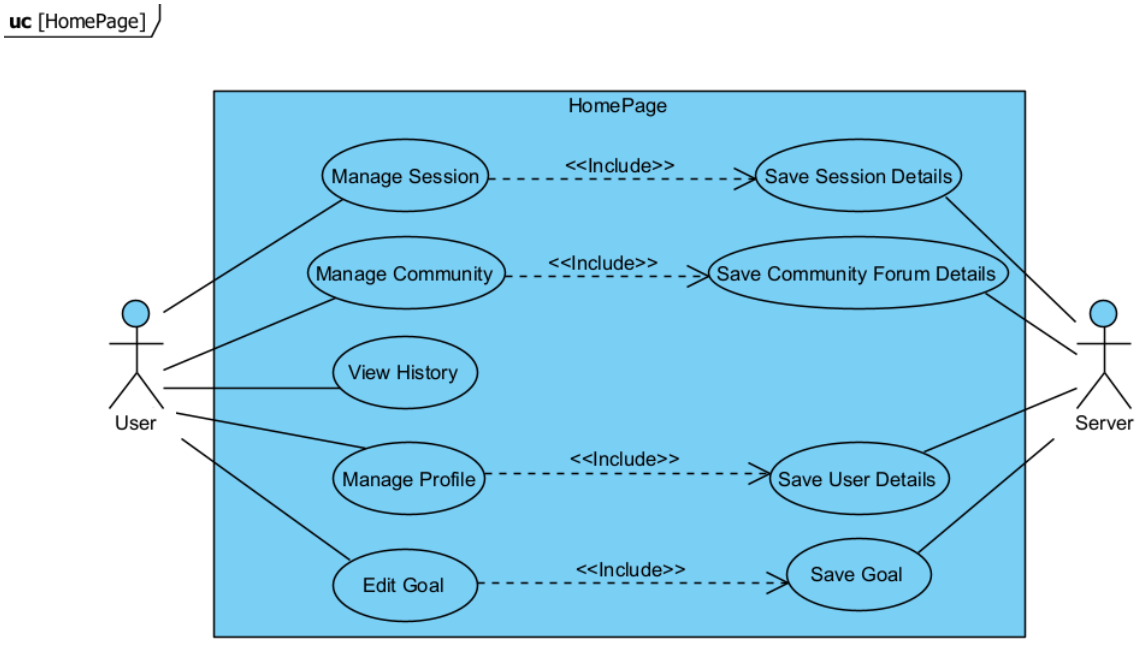
Users are assumed to be connected to the Internet and have their location services enabled.

## 2.4 Use case diagrams for Functionalities

### 2.4.1 Use Case Diagram for Login Page “Urban”




### 2.4.2 Use Case Diagram for Home Page “Urban”




### 2.4.3 Full Use Case Diagram for “Urban”

Urban.pdf

# 3. System Feature

## 3.1 Account Registration

| Use Case ID: | UC001 | | |
|---|---|---|---|
| Use Case Name: | Account Registration | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| Actor: | User, Firebase |
|---|---|
| Description: | Users have to register for an account if they do not possess one to use the functionalities of the application. |
| Preconditions: | User does not have a valid account and wants to register for one. The user must be on the application's start page. |
| Postconditions: | Account details are stored in Firebase and the user will be redirected to the application's home page. |
| Priority: | High |
| Frequency of Use: | Low |
| Flow of Events: | 1. The user presses on the **<Register>** button on the application's start page.<br>2. The system displays the application's registration page.<br>3. The user enters a valid username, first name, last name, password, and email.<br>4. The user submits valid username, first name, last name, password, and email by pressing the <**Sign Up**> button.<br>5. The server validates the username, first name, last name and password.<br>6. The server saves the unique username, first name, last name and password on the Firebase.<br>7. The system displays a successful registration message<br>8. The system displays the application's home screen.<br>9. The use case ends. |
| Alternative Flows: | UC001.AC.1 Missing username, first name, last name, password, or email<br>1. The system prompts the user for the missing inputs.<br>2. Use case resumes at main flow step 3.<br><br>UC001.AC.2 Invalid username, first name, last name, password, or email<br>1. The system prompts the user for the invalid inputs.<br>2. Use case resumes at main flow step 3.<br><br>UC001.AC.3 Username taken<br>1. The system displays a "Username taken" message.<br>2. The system prompts the user for a unique username.<br>3. Use case resumes at main flow step 3. |

| | |
|---|---|
| | UC001.AC.4 Email taken<br>1. The system displays an “Email registered” message.<br>2. The system prompts the user for an unregistered email.<br>3. Use case resumes at main flow step 3.<br><br>UC001.AC.5 Password does not meet requirements<br>1. The system displays a “Password does not meet requirements” message.<br>2. The system prompts the user for a password.<br>3. Use case resumes at main flow step 3. |
| Exceptions: | UC001.EX.1 Cancellation of User Registration<br>1. The user aborts the use case by pressing on the <**Back**> button.<br>2. The system does not save any details.<br>3. The user will be redirected to the application's start page.<br>4. The use case ends. |
| Includes: | SVR.UC001, SVR.UC002 |
| Special Requirements: | |
| Assumptions: | When the user enters the application’s registration page, the user must successfully create an account. |
| Notes and Issues: | |

## 3.2 Login Account

| Use Case ID: | UC002 | | |
|---|---|---|---|
| Use Case Name: | Login Account | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| Actor: | User, Firebase |
|---|---|
| Description: | The user indicates intent to log in to his previously registered account with his registered username and password to access the functionalities of the application. |
| Preconditions: | User has a valid account and is on the application's start page. |
| Postconditions: | User successfully logs in and the application displays the home screen. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. 1. The user presses on the <**Login**> button on the application's start page.<br>2. 2. The system displays the application's login page.<br>3. 3. The user enters a registered username and the associated password.<br>4. 4. The user submits the username and password by pressing the <**Login**> button.<br>5. 5. The server verifies the username and password with Firebase.<br>6. The system displays the application's home screen.<br>7. The use case ends. |
| Alternative Flows: | UC002.AC.01 Missing username and/or passwords<br>1. The system prompts for a username and password.<br>2. The use case resumes at main flow step 3.<br><br>UC002.AC.02 Incorrect username and/or password<br>1. The system displays an "Incorrect username and/or password" message.<br>2. The system prompts for a username and password.<br>3. The use case resumes at main flow step 3. |
| Exceptions: | UC002.EX.1 Cancellation of Account Login<br>1. The user aborts the use case by pressing on the <**Back**> button.<br>2. The system displays the application's start page.<br>3. The use case ends. |
| Includes: | SVR.UC003 |
| Special Requirements: | |
| Assumptions: | When the user enters the application's account login page, the user must successfully login to their registered account. |
| Notes and Issues: | |

## 3.3 Reset Password

| Use Case ID: | UC003 | | |
|---|---|---|---|
| Use Case Name: | Reset Password | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| Actor: | User, Firebase |
|---|---|
| Description: | The user indicates intent to reset their password as he may have forgotten his password. |
| Preconditions: | User has a previously registered account and is on the application's login page. The user remembers his registered email. |
| Postconditions: | User password successfully updated on Firebase and the user will be redirected to the application's login page. |
| Priority: | Low |
| Frequency of Use: | Low |
| Flow of Events: | 1. 1. The user presses on the <**Forgot Password?**> button on the application's login page.<br>2. 2. The system displays the application's reset password page.<br>3. 3. The user enters their email.<br>4. 4. The server sends the user a reset password otp to the email.<br>5. 5.The user inputs the reset password One-Time Password (OTP).<br>6. 6. The user is redirected to the reset password page on the application.<br>7. 7. The user enters a new password.<br>8. 8. The user submits the password.<br>9. 9. The server validates the password.<br>10. 10. The server saves the password on Firebase.<br>11. 11. The system displays the login screen.<br>12. 12. The use case ends. |
| Alternative Flows: | UC003.AC.01 Invalid email<br>1. The system displays an "Invalid email" message.<br>2. The system prompts for a valid email.<br>3. The use case resumes at main flow step 3.<br><br>UC003.AC.02 Invalid password<br>1. The system displays an "Invalid password" message.<br>2. The system prompts for a valid password.<br>3. The use case resumes at main flow step 7.<br><br>UC003.AC.03 Invalid OTP<br>1. The system displays an "Invalid OTP" message.<br>2. The system adds a count to the MaxTries variable.<br>3. The system prompts for a valid OTP.<br>4. The use case resumes at main flow step 5. |
| Exceptions: | UC003.EX.1 Cancellation of Password Reset |

| | |
|---|---|
| | 1. The user aborts the use case.<br>2. The system displays the login page.<br>3. The use case ends.<br><br>UC003.EX.2 Exceed number of MaxTries of OTP allowed<br>1. The user exceeds the number of MaxTries of OTP.<br>2. MaxTries reset to 0 and OTP becomes invalid.<br>3. The system displays the login page.<br>4. The use case ends. |
| Includes: | SVR.UC001, SVR.UC002 |
| Special Requirements: | |
| Assumptions: | When the user enters the application's reset password page, the user must successfully reset their password. |
| Notes and Issues: | |

## 3.4 Manage Profile

| Use Case ID: | UC004 | | |
|---|---|---|---|
| Use Case Name: | Manage Profile | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| Actor: | User, Firebase |
|---|---|
| Description: | The user indicates interest to view his profile. The profile will display his username, first name, last name, email, total distance to date and total hours to date on sessions. The user can edit his profile details. |
| Preconditions: | User is logged in and is on the application's home page. |
| Postconditions: | The user's information is updated on the Firebase, and the user will be redirected to the application's login page. |
| Priority: | Mid |
| Frequency of Use: | Mid |
| Flow of Events: | 1. The user presses the <**Personal**> button on the application's home page.<br>2. The system displays the application's profile page.<br>3. Use case ends. |
| Alternative Flows: | UC004.AC.01 Edit profile<br>1. The user can makinpe changes to his profile by pressing the <**Edit**> icon on the application's profile page.<br>2. The system displays the application's edit profile page.<br>3. Users can update his profile by inputting new profile information.<br>4. User can save his changes by pressing the <**Save>** button.<br>5. User profile information will be updated on Firebase.<br>6. The use case resumes at main flow step 2.<br><br>UC004.AC.02 Cancel edit<br>1. The user can revert any changes to his profile during current edit by pressing the <**Cancel**> button on the application's edit profile page.<br>2. The use case resumes at main flow step 2. |
| Exceptions: | UC004.EX.01 Invalid username, first name, last name and email<br>1. The system will display a "Invalid username, first name, last name and email" message.<br>2. The system will prompt the user to enter a username, first name, last name and email.<br>3. The use case resumes at alt flow UC004.AC.01 step 3. |
| Includes: | SVR.UC002 |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

## 3.5 Manage Session

| Use Case ID: | UC006 | | |
|---|---|---|---|
| Use Case Name: | Manage Session | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| Actor: | User, Firebase, Google Maps API |
|---|---|
| Description: | The user indicates intent to record a session. The user can start, stop, pause and resume a session. The session will show the user's current session details, including his current location on the map, the route from the initial start point of the user to the user's current location on the map, total distance travelled, time taken, and average speed. |
| Preconditions: | User is logged in and is on the application's home page. |
| Postconditions: | The session is stopped and the user's session details are stored on the Firebase. The user will be redirected to the application's home page. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. The user starts a session by either pressing the <**Cycle>** icon or the <**Running>** icon.<br>2. The system will start recording the session.<br>3. The system will constantly update the current session's details.<br>4. The user stops the current session by pressing the <**Stop**> button.<br>5. The system will stop recording the session.<br>6. The system will save the information on the Firebase.<br>7. The system displays the application's home page. |
| Alternative Flows: | UC006.AC.01 Pause and Resume<br>1. The user can pause a current session by pressing the <**Pause**> button either on main flow step 2 or 3.<br>2. The system will temporarily stop updating the session.<br>3. The system will continue updating the session after the user presses the <**Resume>** button.<br>4. The use case resumes at main flow step 3. |
| Exceptions: | |
| Includes: | SVR.UC004 |
| Special Requirements: | |
| Assumptions: | The user will not <**Pause**> the session indefinitely.<br>The user will eventually press the <**Stop**> button. |
| Notes and Issues: | |

## 3.6 Manage Community

| Use Case ID: | UC007 | | |
|---|---|---|---|
| Use Case Name: | Manage Community | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| | |
|---|---|
| Actor: | User, Firebase |
| Description: | The user can interact with other users through the community forum page by posting, liking posts, and commenting under posts. The user can also delete his own posts. The user can end the use case by pressing the **<Back>** button. |
| Preconditions: | User is logged in and is on the application's home page. User has saved at least one session. |
| Postconditions: | The community forum is updated on the Firebase. |
| Priority: | Mid |
| Frequency of Use: | High |
| Flow of Events: | 1. The user selects the Community screen by pressing the <**Community**> icon.<br>2. The user chooses to create a post by pressing the **<Post>** button.<br>3. The user chooses a session to post about.<br>4. The user enters a description for the post.<br>1. 5. The user submits the post.<br>2. 6. The post will be saved by the Firebase.<br>3. 6. The community forum will be updated with the user's post.<br>4. 7. The use case ends. |
| Alternative Flows: | UC007.AC.01 Post Deleted<br>1. The user searches for the post they created to delete after main flow step 1.<br>2. The user chooses the post for deletion by pressing the **<Delete>** button beside the post.<br>3. The system prompts the user for confirmation of deletion.<br>4. The user confirms.<br>5. The post is deleted from the Firebase and the community forum will be updated.<br>6. The use case ends.<br><br>UC007.AC.02 Like Post<br>1. The user searches for a post to like after main flow step 1.<br>2. The user chooses a post to like by pressing the **<Like>** icon beside the post.<br>3. The user likes the post.<br>4. The post is updated on the Firebase and the community forum will be updated.<br>5. The use case ends.<br><br>UC007.AC.03 Unlike Post<br>1. The user searches for a post they liked to unlike after main flow step 1. |

| | |
|---|---|
| | 2. The user chooses the post to unlike by pressing the **<Like>** icon beside the post.<br>3. The user unlikes the post.<br>4. The post is updated on the Firebase and the community forum will be updated.<br>5. The use case ends.<br><br>UC007.AC.04 Comment under post<br>1. The user searches for a post to comment under after main flow step 1.<br>2. The user chooses a post to comment under by pressing the **<Comment>** button on a post.<br>3. The user enters a comment.<br>4. The user submits the comment.<br>5. The post is updated on the Firebase and the community forum will be updated.<br>6. The use case ends.<br><br>UC007.AC.05 Delete comment under post<br>1. The user searches for the comment they created to delete after main flow step 1.<br>2. The user chooses the comment for deletion by pressing the **<Delete>** button beside the comment.<br>3. The system prompts the user for confirmation of deletion.<br>4. The user confirms the deletion.<br>5. The post is updated on the Firebase and the community forum will be updated.<br>6. The use case ends.<br><br>UC007.AC.06 Abort Manage Community<br>1. The user aborts the use case by pressing on the <**Back**> button.<br>2. The user will be directed to the application's home page.<br>3. The use case ends. |
| Exceptions: | |
| Includes: | SVR.UC005 |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

## 3.7 View History

| Use Case ID: | UC008 | | |
|---|---|---|---|
| Use Case Name: | View History | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| Actor: | User, Firebase |
|---|---|
| Description: | The user indicates the intent to view details about his past sessions. The details include the date, start time, distance travelled, timing, average speed and cycling route of the session. |
| Preconditions: | User is logged in, has past sessions and is on the application's home page. |
| Postconditions: | The system displays the relevant information about the user's past sessions. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. 1. The user can view his past sessions by pressing the <**Sessions**> button on the application's home page.<br>2. 2. The system displays the application's past sessions which were saved in Firebase.<br>3. 3. The user can press on the sessions to view the route of the session, traced on a map.<br>4. 4. The use case ends. |
| Alternative Flows: | |
| Exceptions: | |
| Includes: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

## 3.8 Edit Goal

| Use Case ID: | UC009 | | |
|---|---|---|---|
| Use Case Name: | Edit Goal | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| Actor: | User, Firebase |
|---|---|
| Description: | The monthly goal will be displayed on the top of the application's session page with a default goal of 50 kilometres. The user indicates the intent to set a new monthly goal and can edit the monthly goal indefinitely. The new monthly goal will be displayed. The goal counter will start with a value of 0 and is the sum of distance travelled during the sessions recorded in the current month. Goal counter will reset at the start of every month. |
| Preconditions: | User is logged in and is on the application's session page. |
| Postconditions: | The monthly goal is successfully updated. |
| Priority: | Low |
| Frequency of Use: | Mid |
| Flow of Events: | 5. 1. The user can edit a monthly goal by pressing the <**Monthly Goal**> icon on the application's sessions page.<br>6. 2. The user inputs the monthly goal (in kilometres).<br>7. 3. The user submits the monthly goal by pressing the <**Confirm**> button.<br>8. 4. The system validates the monthly goal.<br>9. 5. The goal is updated on the Firebase.<br>10. 6. The system successfully displays the new monthly goal.<br>11. 7. The use case ends. |
| Alternative Flows: | |
| Exceptions: | UC009.EX.1 Invalid Monthly goal<br>1. The system displays "Monthly goal must not exceed 999KM" message.<br>2. The system prompts the user to enter a monthly goal.<br>3. The use case resumes at main flow step 2.<br><br>UC009.EX.2 Cancellation of Goal Setting<br>1. User aborts the use case by pressing the <**X**> icon.<br>2. The system displays the application's session page.<br>3. The use case ends. |
| Includes: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

## 3.9 SVR.UC001 Validate User Details

| Use Case ID: | SVR.UC001 | | |
|---|---|---|---|
| Use Case Name: | Validate User Details | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| | |
|---|---|
| Actor: | Server, Firebase |
| Description: | The use case begins when the server receives user details to validate. The use case ends when the username is not taken, email not taken, password meets requirements and first name/last name has been filled. |
| Preconditions: | Email entered, password entered, username entered, first name entered and last name entered. |
| Postconditions: | Accurately verified the user details and sends an appropriate message to the system. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. 1. The server receives user details.<br>2. 2. The server validates that all user details are not NULL.<br>3. 2. The server validates the email is not used and in valid format<br>4. 3. The server validates the unique username is unique.<br>5. 4. The server validates the first name is not missing.<br>6. 5. The server validates the last name is not missing.<br>7. 6. The server validates the password meets the requirements.<br>8. 7. The server successfully validates all the user details.<br>9. 8. The use case ends. |
| Alternative Flows: | SVR.UC001.AC1 Username Taken.<br>1. 1. The server sends a “Username Taken” message to the system.<br>2. 2. The use case ends.<br><br>SVR.UC001.AC2 Email Taken<br>3. 1. The server sends an “Email Taken” message to the system.<br>4. 2. The use case ends.<br><br>SVR.UC001.AC3 First Name and/or Last Name is missing<br>5. 1. The server sends a “First Name and/or Last Name is missing” message to the system.<br>6. 2. The use case ends.<br><br>SVR.UC001.AC4 Password does not meet requirements<br>7. 1. The server sends a “Password does not meet requirements” message to the system.<br>8. 2. The use case ends.<br><br>SVR.UC001.AC5 Empty inputs |

| | |
|---|---|
| | 1. The server sends a “Missing inputs” message to the system.<br>2. The user case ends. |
| Exceptions: | |
| Includes: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

## 3.10 SVR.UC002 Save User details

| Use Case ID: | SVR.UC002 | | |
|---|---|---|---|
| Use Case Name: | Save User details | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| | |
|---|---|
| Actor: | Server, Firebase |
| Description: | The use case begins when the actor receives validated user details - email, username, password, first name and last name. |
| Preconditions: | Email validated, password validated, username validated, first name validated, and last name validated. |
| Postconditions: | User details are saved successfully on Firebase. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. 1. The server receives validated user details.<br>2. 2. The server saves the email on Firebase.<br>3. 3. The server saves the username on Firebase.<br>4. 4. The server saves the first name on Firebase.<br>5. 5. The server saves the last name on Firebase.<br>6. 6. The server saves the password on Firebase.<br>7. 7. The server successfully saved all user details on Firebase.<br>8. 8. The use case ends. |
| Alternative Flows: | |
| Exceptions: | SVR.UC002.EX1 Storage space full<br>1. The server aborts the case due to not enough space left in storage.<br>2. The server sends a “Insufficient Storage, please contact an admin” message to the system.<br>3. The use case ends. |
| Includes: | |
| Special Requirements: | |
| Assumptions: | The user details will always be successfully saved after validation of details. |
| Notes and Issues: | |

## 3.11 SVR.UC003 Verify Login details

| Use Case ID: | SVR.UC003 | | |
|---|---|---|---|
| Use Case Name: | Verify Login details | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| | |
|---|---|
| Actor: | Server, Firebase |
| Description: | The use case begins when the actor receives user details to verify and ends when it has been verified with Firebase. |
| Preconditions: | Valid Username and associated password entered. |
| Postconditions: | Accurately verifies the authenticity of username and password pair with Firebase and server sends an appropriate message to the system. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. 1 .The server receives a username and password pair.<br>2. 2. The server verifies the username is in Firebase and the password received matches with the password stored in Firebase.<br>3. 3. The server sends a “Verified” message to the system.<br>4. 4. The use case ends. |
| Alternative Flows: | SVR.UC003.AC1 Username not found and/or Password mismatch<br>1. The server sends a “Username not found and/or Password mismatch” message to the system.<br>2. The use case ends. |
| Exceptions: | |
| Includes: | |
| Special Requirements: | |
| Assumptions: | |
| Notes and Issues: | |

## 3.12 SVR.UC004 Save Session details

| Use Case ID: | SVR.UC004 | | |
|---|---|---|---|
| Use Case Name: | Save Session details | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| Actor: | Server, Firebase |
|---|---|
| Description: | The use case begins when the server receives session details - Route, Total distance travelled, time taken and average speed. |
| Preconditions: | User is logged in and has stopped a session on the session page. |
| Postconditions: | User session details are saved successfully on Firebase. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 1. 1. The server receives user session details.<br>2. 2. The server saves the route on Firebase.<br>3. 3. The server saves the total distance travelled on Firebase.<br>4. 4. The server saves the time taken on Firebase.<br>5. 5. The server saves the average speed on Firebase.<br>6. 6. The use case ends. |
| Alternative Flows: | |
| Exceptions: | SVR.UC004.EX1 Storage space full<br>1. The server aborts the case due to not enough space left in storage.<br>2. The server sends a “Insufficient Storage, please contact an admin” message to the system.<br>3. The use case ends. |
| Includes: | |
| Special Requirements: | |
| Assumptions: | The user details will always be successfully saved. |
| Notes and Issues: | |

## 3.13 SVR.UC005 Save Community Forum details

| Use Case ID: | SVR.UC005 | | |
|---|---|---|---|
| Use Case Name: | Save Community Forum details | | |
| Created By: | Arun Ezekiel | Last Updated By: | Lee Wei Xian |
| Date Created: | 01/02/2023 | Date Last Updated: | 14/02/2023 |

| Actor: | Server, Firebase |
|---|---|
| Description: | The use case begins after the user interacts with the community forum - Post created, post deleted, post liked, post unliked, post commented, comment deleted. |
| Preconditions: | User is logged in and interacts with the community forum. |
| Postconditions: | Community forum details are saved successfully on Firebase. |
| Priority: | High |
| Frequency of Use: | High |
| Flow of Events: | 7. 1. The server receives community forum interaction.<br>8. 2. The server saves the appropriate details on Firebase.<br>9. 3. The use case ends. |
| Alternative Flows: | |
| Exceptions: | SVR.UC005.EX1 Storage space full<br>1. The server aborts the case due to not enough space left in storage.<br>2. The server sends a “Insufficient Storage, please contact an admin” message to the system.<br>3. The use case ends. |
| Includes: | |
| Special Requirements: | |
| Assumptions: | The community forum details will always be successfully saved. |
| Notes and Issues: | |