Software Requirements Specification

for

<Urban>

Version 1.1 approved

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

<SC2001 Software Engineering Z59_Group1>

<22/03/2023>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Lee Wei Xian	22/3/2023	Updated Community Functionality Minor edit in flow of use cases	1.1

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. Additionally, the application allows users to view session history, edit goals and edit personal profiles.

2. Overall Description

2.1 Product Perspective

“Urban” integrates Google Maps API to provide up-to-date user position on the map, allowing for tracking of user’s distance during a session. The application also leverages Google Firebase, by storing and displaying user data such as past sessions, goals and their 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
- Display map

Minor Functions:

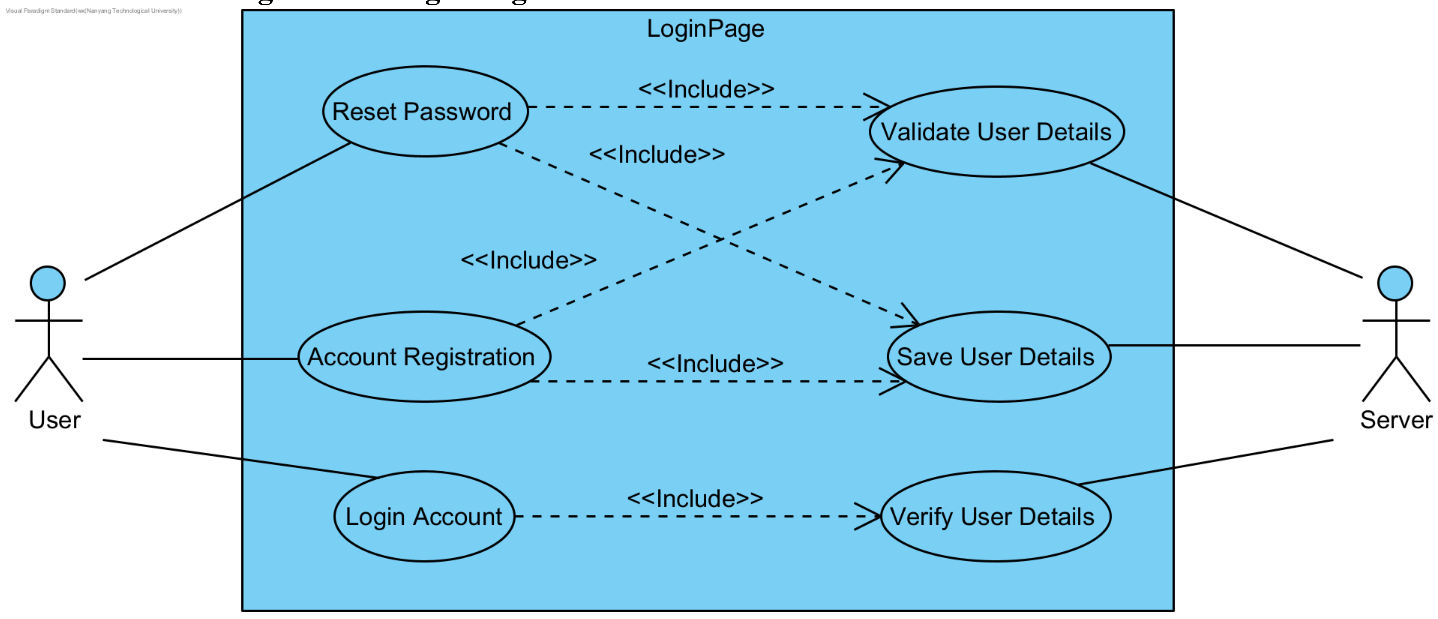
- Reset forgotten password
- Edit goal
- Manage profile
- Pause/Resume live session

2.3 Assumptions and Dependencies

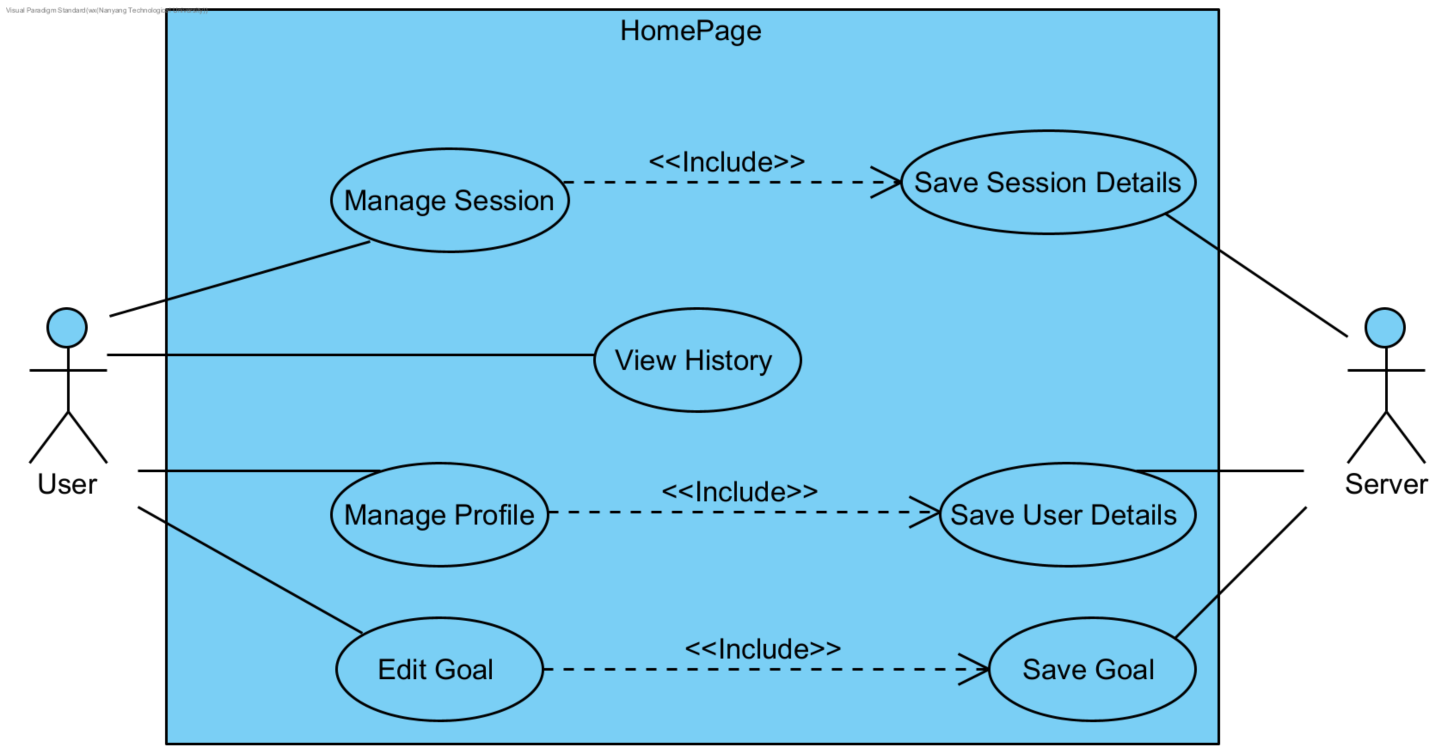
Users are assumed to be connected to the Internet and have their location services enabled.

2.4 Use case diagrams for Functionalities

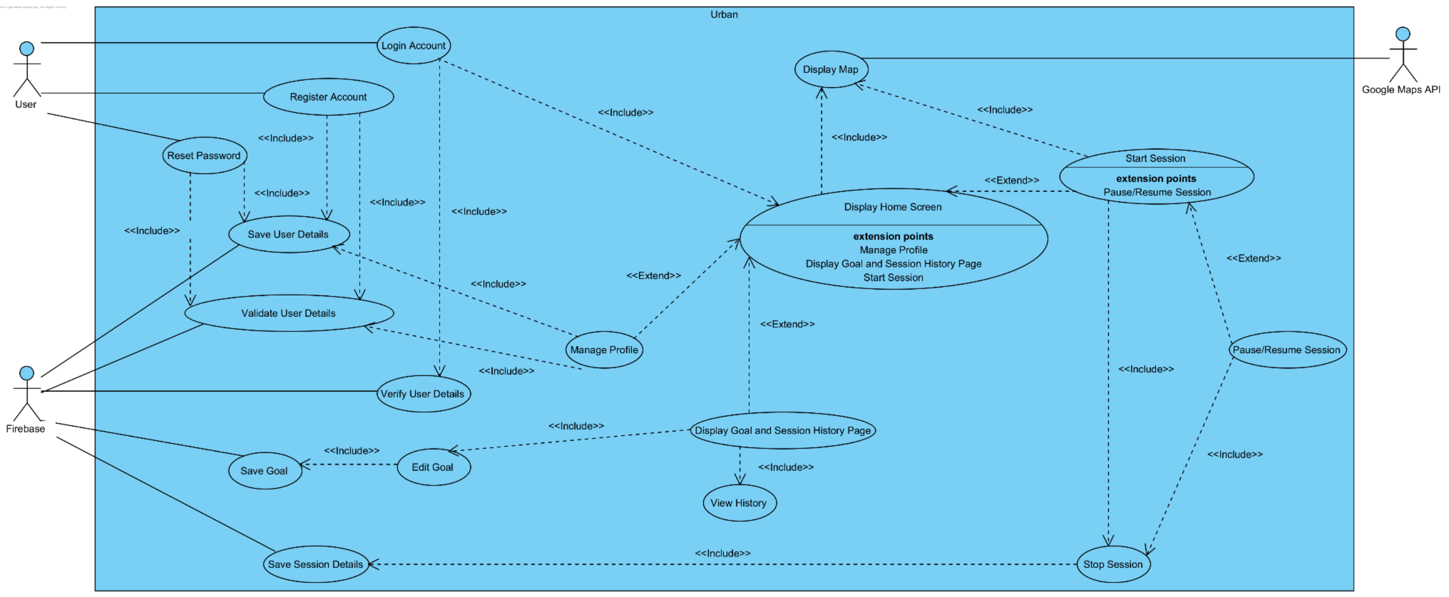
2.4.1 Use Case Diagram for Login Page "Urban"



2.4.2 Use Case Diagram for Home Page "Urban"



2.4.3 Full Use Case Diagram for "Urban"


Urban
User
Login Account
Register Account
Reset Password
Save User Details
Validate User Details
Verify User Details
Firebase
Save Goal
Edit Goal
Save Session Details
Manage Profile
Display Map
Google Maps API
Display Home Screen
extension points
Manage Profile
Display Goal and Session History Page
Start Session
Start Session
extension points
Pause/Resume Session
Pause/Resume Session
Display Goal and Session History Page
View History
Stop Session
<<Include>>
<<Extend>>

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:	22/03/2023

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 is created in Firebase with input details. 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. 2. The system displays the application’s registration page. 3. The user enters a username, first name, last name, password, and a unique email. 4. The user submits valid username, first name, last name, password, and email by pressing the <**Sign Up**> button. 5. The server validates the input information and sends a request to Firebase. 6. Firebase ensures the email input is unique. 6. Firebase saves the username, first name, last name, password, and email. 7. The system displays a successful registration message. 8. The system displays the application’s home screen. 9. The use case ends.
Alternative Flows:	UC001.AC.1 Invalid username, first name, last name, password, or email 1. The system prompts the user for the invalid inputs. 2. Use case resumes at main flow step 3. UC001.AC.2 Email taken 1. The system displays an “Email registered” message. 2. The system prompts the user for an unregistered email. 3. Use case resumes at main flow step 3.
Exceptions:	UC001.EX.1 Cancellation of User Registration

	1. The user aborts the use case by pressing on the <**Back>** button. 2. The system does not save any details. 3. The user will be redirected to the application's start page. 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:	23/03/2023

Actor:	User, Firebase
Description:	The user indicates intent to log in to his previously registered account with his registered email and password to access the functionalities of the application.
Preconditions:	User has previously created a valid account (stored in Firebase). User is on the application's start page.
Postconditions:	User successfully logs in. The user is redirected to the application's 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. 2. 2. The system displays the application's login page. 3. 3. The user enters a registered email and the associated password. 4. 4. The user submits the email and password by pressing the <**Login**> button and sends a request to the Firebase. 5. 5. Firebase verifies the email and password. 6. The system displays the application's home screen. 7. The use case ends.
Alternative Flows:	UC002.AC.01 Missing email and/or password 1. The system prompts for missing email and/or password. 2. The use case resumes at main flow step 3. UC002.AC.02 Incorrect email and/or password 1. The system displays an "Incorrect email and/or password" message. 2. The system prompts for an email and password. 3. The use case resumes at main flow step 3.
Exceptions:	UC002.EX.1 Cancellation of Account Login 1. The user aborts the use case by pressing on the <**Back**> button. 2. The system displays the application's start page. 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:	23/03/2023

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. User is on the application's login page. The user is able to access his registered email.
Postconditions:	User password successfully updated on Firebase. Users 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. 2. 2. The system displays the application's reset password page. 3. 3. The user enters their account's email. 4. 4. A request is sent to Firebase. 5. 5. Firebase validates the email address. 6. 6. The server sends the user a reset password link via email. 7. 7. The user enters the link and is redirected to the reset password page on the application. 8. 7. The user enters a new password. 9. 8. The user submits the password. 10. 9. The server validates the password and sends a request to Firebase. 11. 10. Firebase updates the account password of the user. 12. 11. The system displays the login screen. 13. 12. The use case ends.
Alternative Flows:	UC003.AC.01 Invalid email 1. The system displays an "Invalid email" message. 2. The system prompts for a valid email. 3. The use case resumes at main flow step 3. UC003.AC.02 Invalid password 1. The system displays an "Invalid password" message. 2. The system prompts for a valid password. 3. The use case resumes at main flow step 7.
Exceptions:	UC003.EX.1 Cancellation of Password Reset 1. The user aborts the use case. 2. The system displays the login page. 3. 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:	23/03/2023

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. User is on the application's home page.
Postconditions:	The user's information is updated on the Firebase. Users 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. 2. The system displays the application's profile page. 3. Use case ends.
Alternative Flows:	UC004.AC.01 Edit profile 1. The user can make changes to his profile by pressing the <**Edit**> icon on the application's profile page. 2. The system displays the application's edit profile page. 3. Users can update his profile by inputting a new username, first name, or last name. 4. User can save his changes by pressing the <**Save>** button. 5. The server sends a request to Firebase. 6. Firebase will update the user profile information. 7. The use case resumes at main flow step 2. UC004.AC.02 Cancel edit 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. 2. The use case resumes at main flow step 2.
Exceptions:	UC004.EX.01 Invalid username, first name and/or last name 1. The system will display a "Invalid username, first name and/or last name" message. 2. The system will prompt the user to enter a new username, first name and/or last name. 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:	UC005		
Use Case Name:	Manage Session		
Created By:	Arun Ezekiel	Last Updated By:	Lee Wei Xian
Date Created:	01/02/2023	Date Last Updated:	23/03/2023

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 session to the user’s current location on the map, total distance travelled, time taken, and average speed.
Preconditions:	User is logged in. User 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 <**Run>** icon. 2. The system will start recording the session. 3. The system will constantly update the current session’s details. 4. The user stops the current session by pressing the <**Stop**> button. 5. The system will stop recording the session and send a request to Firebase. 6. Firebase will save the session with the user’s id and session’s details. 7. The system displays the application’s home page.
Alternative Flows:	UC005.AC.01 Pause and Resume 1. The user can pause a current session by pressing the <**Pause**> button either on main flow step 2 or 3. 2. The system will temporarily stop updating the session. 3. The system will continue updating the session after the user presses the <**Resume>** button. 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. The user will eventually press the <**Stop**> button.
Notes and Issues:	

3.6 View History

Use Case ID:	UC006		
Use Case Name:	View History		
Created By:	Arun Ezekiel	Last Updated By:	Lee Wei Xian
Date Created:	01/02/2023	Date Last Updated:	23/03/2023

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. User has past sessions. User 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. 2. 2. The system displays the application's past sessions which were saved in Firebase. 3. 3. The user can press on the sessions to view past session details. 4. 4. The system will display the date, start time, distance travelled, timing, average speed and cycling route of the session. 5. 5. The use case ends.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.7 Edit Goal

Use Case ID:	UC007		
Use Case Name:	Edit Goal		
Created By:	Arun Ezekiel	Last Updated By:	Lee Wei Xian
Date Created:	01/02/2023	Date Last Updated:	23/03/2023

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 goal counter will start with a value of 0 and is the sum of distance travelled during the sessions recorded in the current month. The user indicates the intent to set a new monthly goal.
Preconditions:	User is logged in. User is on the application's session page.
Postconditions:	The monthly goal is successfully updated on the Firebase. The monthly goal is shown on the session page.
Priority:	Low
Frequency of Use:	Mid
Flow of Events:	6. 1. The user can edit a monthly goal by pressing the <**Monthly Goal>** icon on the application's sessions page. 7. 2. The user inputs the monthly goal (in kilometres). 8. 3. The user submits the monthly goal by pressing the <**Confirm**> button. 9. 4. The system validates the monthly goal and sends a request to Firebase. 10. 5. The user goal is updated on the Firebase. 11. 6. The system successfully displays the new monthly goal. 12. 7. The use case ends.
Alternative Flows:	
Exceptions:	UC007.EX.1 Invalid Monthly goal exceeding 999KM 1. The system displays "Monthly goal must not exceed 999KM" message. 2. The system prompts the user to enter a valid monthly goal. 3. The use case resumes at main flow step 2. UC007.EX.2 Cancellation of Goal Setting 1. User aborts the use case by pressing the <**X**> icon. 2. The system displays the application's session page. 3. The use case ends.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.8 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:	23/03/23

Actor:	Server, Firebase
Description:	The use case begins when the server receives user details to validate. The use case ends when the input email is unique, while username, first name, last name, and password is filled and valid.
Preconditions:	Email, username, first name, last name, and password 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. 2. 2. The server validates that all user details, including email, username, first name, last name, and password, are not missing and is valid. 3. 3. The server sends a request to Firebase to validate input email. 4. 2. The Firebase validates the email is not used and in valid format 5. 7. The server successfully validates all the user details. 6. 8. The use case ends.
Alternative Flows:	1. SVR.UC001.AC1 Email Taken 2. 1. The server sends an “Email Taken” message to the system. 3. 2. The use case ends. SVR.UC001.AC2 Empty inputs 1. The server sends a “Missing inputs” message to the system. 2. The user case ends.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.9 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:	23/03/23

Actor:	Server, Firebase
Description:	The use case begins when the actor receives validated user details - email, username, password, first name and last name.
Preconditions:	User details, including email, password, username, first name, 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 sends a request to Firebase with validated user details. 2. 2. The server saves the email on Firebase as a key. 3. 3. The server saves the username, first name, last name and password on Firebase. 4. 4. The server successfully saved all user details on Firebase. 5. 5. The use case ends.
Alternative Flows:	
Exceptions:	SVR.UC002.EX1 Storage space full 1. The server aborts the case due to not enough space left in storage. 2. The server sends a “Insufficient Storage, please contact an admin” message to the system. 3. The use case ends.
Includes:	
Special Requirements:	
Assumptions:	The user details will always be successfully saved after validation of details.
Notes and Issues:	

3.10 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:	23/03/2023

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 email and associated password entered in the login page.
Postconditions:	Accurately verifies the authenticity of email 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 an email and password pair. 2. 2. The server verifies the email is in Firebase and the password received matches with the password stored in Firebase. 3. 3. The server sends a "Verified" message to the system. 4. 4. The use case ends.
Alternative Flows:	SVR.UC003.AC1 Email not found 1. The server sends a "Email not found" message to the system. 2. The use case ends. SVR.UC003.AC2 Password mismatch 1. The server sends a "Password mismatch" message to the system. 2. The use case ends.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

3.11 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:	23/03/2023

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. User 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. 2. 2. The server saves the session details, including user id, route, total distance travelled, time taken, and average speed on Firebase. 3. 3. The use case ends.
Alternative Flows:	
Exceptions:	SVR.UC004.EX1 Storage space full 1. The server aborts the case due to not enough space left in storage. 2. The server sends a “Insufficient Storage, please contact an admin” message to the system. 3. The use case ends.
Includes:	
Special Requirements:	
Assumptions:	The user details will always be successfully saved.
Notes and Issues:	