Software Requirements Specification

for

TrailX

Version 2.0 approved.

Prepared by

Team Atom

Asuri Simhakutty Kamakshi (U1923890E)

Kesarimangalam Srinivasan Abhinaya (U1923302J)

Musthiri Sajna (U1922559D)

Singh Aishwarya(U1923952C)

Unnikrishnan Malavika (U1923322E)

17/04/2021

Table of Contents

Revision History

Name	**Date**	**Reason for Changes**	**Version**
All	04/04/21	Add all diagrams and written documentation.	1
Unnikrishnan Malavika	05/04/21	Update functional requirements to include change in password criteria.	1.1
Kesarimangalam Srinivasan Abhinaya	11/04/21	Update changes in use case descriptions, use case diagrams and dialog map	1.2
Unnikrishnan Malavika	11/04/21	Update changes in sequence diagrams and Data dictionary	1.3
Musthiri Sajna	15/04/21	Update changes in user interfaces Update fundamental requirements of Music and business rules	1.4
Asuri Simhakutty Kamakshi	17/04/21	Updated SOLID principles as well as class diagrams	1.5
Singh Aishwarya	17/04/21	Complete document formatting	2.0

1. Introduction

1.1 Purpose

The purpose of this document is to provide an insight into the features and functions of *TrailX*, version 1.0, an android based mobile application that allows its user base to explore and discover new walking trails in Singapore.

1.2 Document Conventions

- The headings of segments in this document are in the font Times New Roman and font size 18 for the main heading. The subsequent headings are in the font Times New Roman, with a decrease in font size by 4 and then subsequently by 2 from the font size just above the heading.
- All the headings are **bolded**.
- The contents in this document are in the font Times New Roman with font size 12.
- Every paragraph is 1.5 line spaced for readability.
- Nested or indented numbering in the functional requirements represents the hierarchy of requirements.
- All mobile application screens have been underlined.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers and testers of the application. It is also meant for the course supervisors of CE2006 to attain a detailed understanding of our project. Readers are recommended to follow the sequential order starting from the Introduction, consequently to the rest of the document till the appendix. For a clear understanding of the document, functional and non-functional requirements, external interfaces, system features and supporting diagrams need to be considered. For developers and testers references, use case diagrams, use case descriptions, class diagrams, sequential diagrams, dialog maps and data dictionary are provided.

1.4 Product Scope

TrailX is a mobile application that motivates its users to lead a healthier lifestyle by enabling them to explore new walking trails in Singapore to reconnect and restore our crumbling relationship with nature. Through *TrailX*, users can rediscover Singapore through the plethora of trails. *TrailX* will be considered complete when the application has been tested and approved for release by the SEA Organization. This project supports the smart nation goal of smart urban mobility to support their vision.

2. Overall Description

2.1 Product Perspective

The *TrailX* application is a self-contained open-source application. The product is developed and deployed on the Android Studio Platform. The main objective of this application is for users living in Singapore to take walking trails in different locations of the country to explore and relax. There are two different categories of trails namely, city trails and nature trails. The users can keep a step count and save the trails that they take.

2.2 Product Functions

The following are the main features that are included in *TrailX*:

- **Sign-in System**: The system allows for the user to create an account with the application using their email address. A User profile is also created by the user with relevant information. This profile as well as the email address and password can be updated by the user in the Settings section of the app.
- **My Account**: The user can view their profile information as well as their trails statistics.
- **Active Trail**: The Active trail that the user is currently talking will be displayed here. A map of the route, user’s live location, the weather information, steps completed, and calories burnt while on the trail will be displayed on the page here. The trail can be paused via this screen.
- **My Trail**: The user’s saved trails will be displayed here.
- **Discover New Trails**: The user can view the available trails by type and trails by distance based on their location on this screen.

- **Music System**: The user can choose from in – app music or navigating to an external music application namely YouTube.

2.3 User Classes and Characteristics

There is only one kind of user class for our *TrailX* application. This user class holds the following user information - full name, email address, password, age, height, weight, and username. This user class is utilized whenever the application runs, to extract user information from the database. The user interface is straightforward and hence the user does not require any experience to utilize the application. Although, the user is expected to know the basic utilization techniques of a smartphone mobile application to seamlessly move through the screens on *TrailX*.

2.4 Operating Environment

The application is built on Android Studio. Therefore, it is compatible with all operating systems that are built on the Android operating system. The application is built with a minimum API level of 22. The application requires access to the internet, location access as well as physical activity access and will be able to function seamlessly in the existence of the above.

2.5 Design and Implementation Constraints

The application requires Google Play services to be present to communicate with the database hosted at Firebase. Therefore, the user will be unable to utilize the application in the absence of the same. Internet access is required by the application. Therefore, the application will be unable to run on a device not connected to Wi-Fi or a data network.

Firebase stores information in a NoSQL format. The communication between the application and the Weather API utilizes the HTTP protocol. For security purposes, Firebase's own authentication method has been incorporated into the application rather than building an authentication system from scratch. The application uses Kotlin as the main programming language.

2.6 User Documentation

The users would be provided with a simple guide through the user interface and the features of the application as a part of the application description when the application is published to the Play Store.

No separate user documentation would be released as the user interface is simple with a small learning curve for the various features that have been integrated into the application.

2.7 Assumptions and Dependencies

The following assumptions are made during the construction of the application and are required for the smooth functioning of the application:

- The application is running on a smartphone that is currently running an Android 11.
- The user is connected to the Internet.
- Location access permission is granted when the application requests for the same to display live location of the user during the trail.
- Physical activity permission is granted when the application requests for the same to display step count of the user during the trail.

In addition to this, the following dependencies were included in the production of *TrailX*:

- Android YouTube player – Version 10.0.5 – dependency for the music function
- The Kotlin standard library for backend programming of the application
- Androidx: AppCompat – Version 1.2.0 – Contains APIs for better user interface experience.
- Android Material – Version 1.3.0 – Dependency for user interface
- Android Constraint Layout – Version 2.0.4 – Dependency for user interface
- Google Play Services Maps – Version 17.0.0 – Used for rendering map layout in Android Studio.
- Mapbox SDK – Version 8.5.1 – Dependency for using map functions, displaying routes, and accessing location points on map.
- Google Play Services Location – Version 18.0.0 – Dependency for accessing user location on Android.
- Firebase – Version 26.7.0 – Dependency for Firebase Authentication Library.

3. External Interface Requirements

3.1 User Interfaces

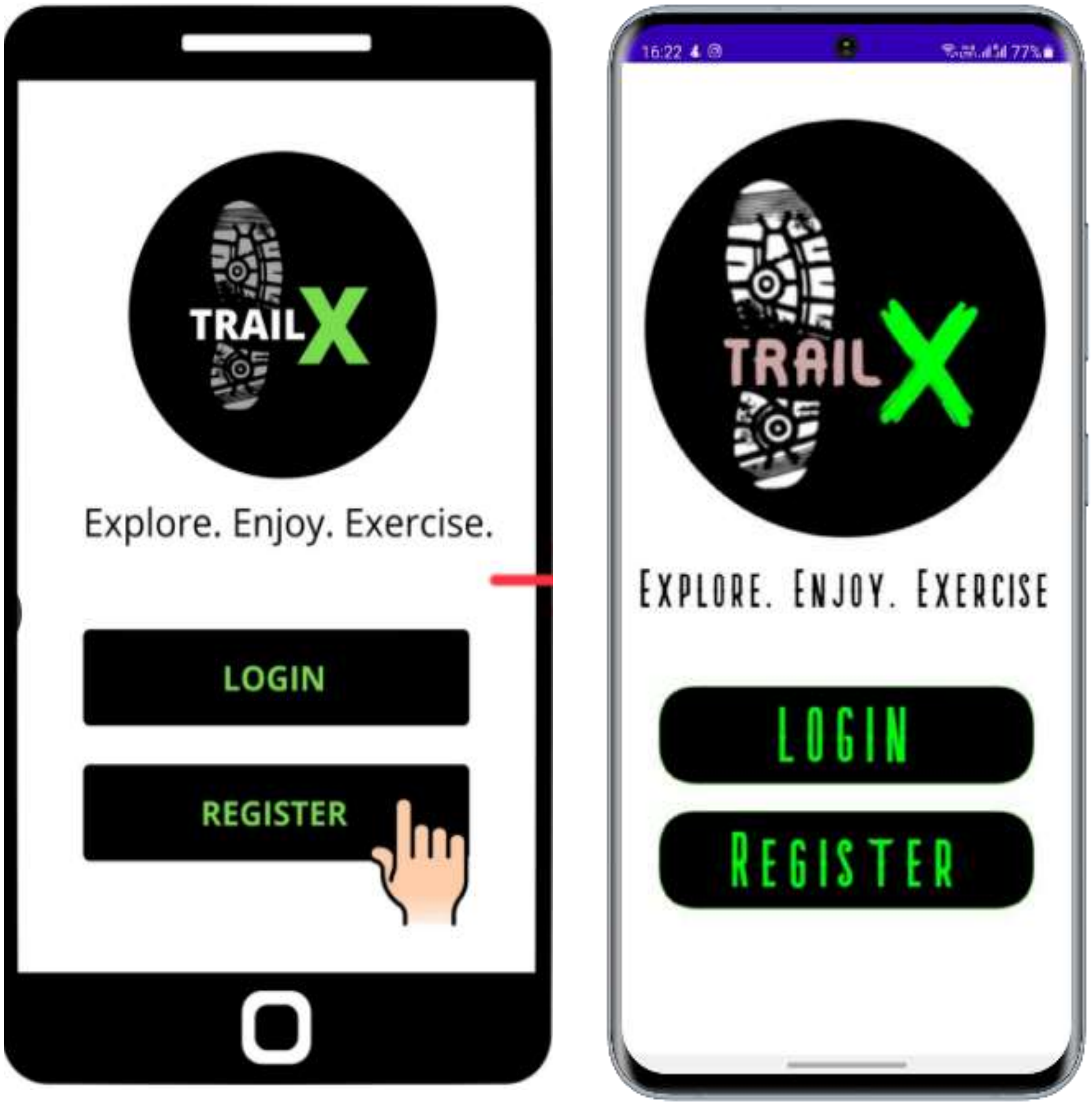
TrailX is a highly interactive application and has a simple and consistent layout. The interactive features include buttons, commonly used icons, search, error messages, icon bars and pop-up messages that ask for user permissions. Every page consists of buttons with familiar images for users' convenience and consistency. When user enters invalid information, error messages appear to prevent errors. The application also consists of dialogue boxes that appear to inform and request permission from the users when accessing their details like live location. These functions provide feedback to the user. By these features, users are aware of the outcomes to their actions. The icon bar in *TrailX* application consists only five items/functions which is within the limit of human attention, thus reducing short term memory load.

Every function in the application is a result of the users' action. The screen layouts in the application that have a user interface are the Login Screen, Register Screen, Home Screen, Active Trail Screen, Start Screen, My Account Screen, Settings Screen, Music Screen, In App Music Screen, Emergency SOS Screen and My Trails Screen. These screen layouts require user interaction to execute further main actions, which will be elaborated upon in this section of the document. For ease of use, the current user interface of the screens has been displayed alongside the user interface mockups that were submitted in the early stages of the mobile application planning and production.

Screen 1: Start Screen

The Start Screen has two buttons “Login” and “Register” which redirect the user to the Login Screen or Register Screen.

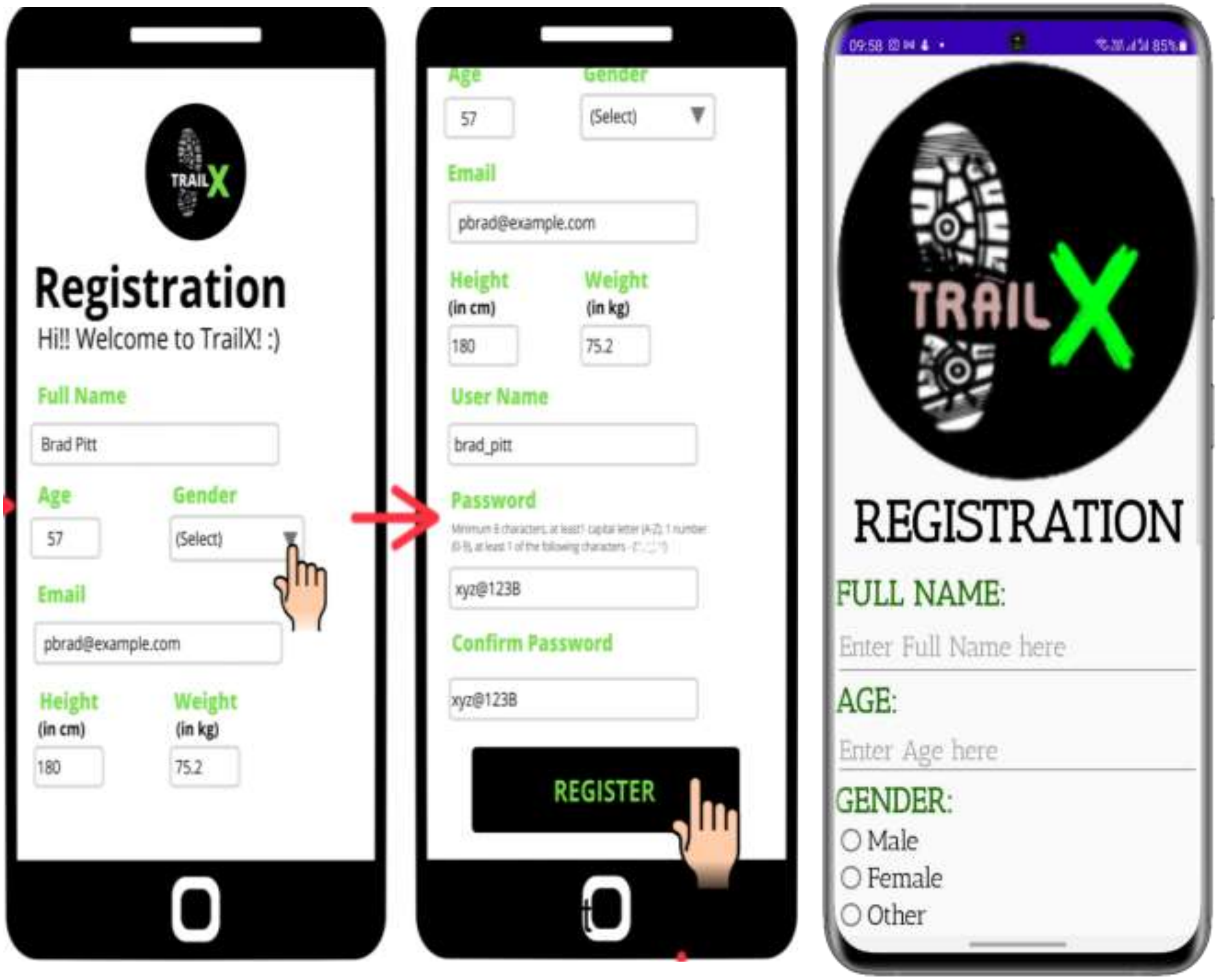
Figure 1: Start Screen: User Interface Mockup (left) versus final User Interface layout (right)



Screen 2: Register Screen

The Register Screen has simple input fields for full name, age, gender, email, height (in centimeters), weight (in kilograms), username and password that obtain the new user data and register the users. After the input fields are filled with correct data, validated in the backend by error checks, then clicking the "Register" button navigates the user to the Home Screen.

Figure 2: Register Screen: User Interface Mockup (left) versus final User Interface layout (right)



Registration

Hi!! Welcome to TrailX! :)

Full Name: Brad Pitt

Age: 57

Gender: (Select)

Email: pbrad@example.com

Height (in cm): 180

Weight (in kg): 75.2

User Name: brad_pitt

Password: xyz@123B

Minimum 8 characters, at least 1 capital letter (A-Z), 1 number (0-9), at least 1 of the following characters -([illegible])

Confirm Password: xyz@123B

REGISTRATION

FULL NAME: Enter Full Name here

AGE: Enter Age here

GENDER:

○ Male

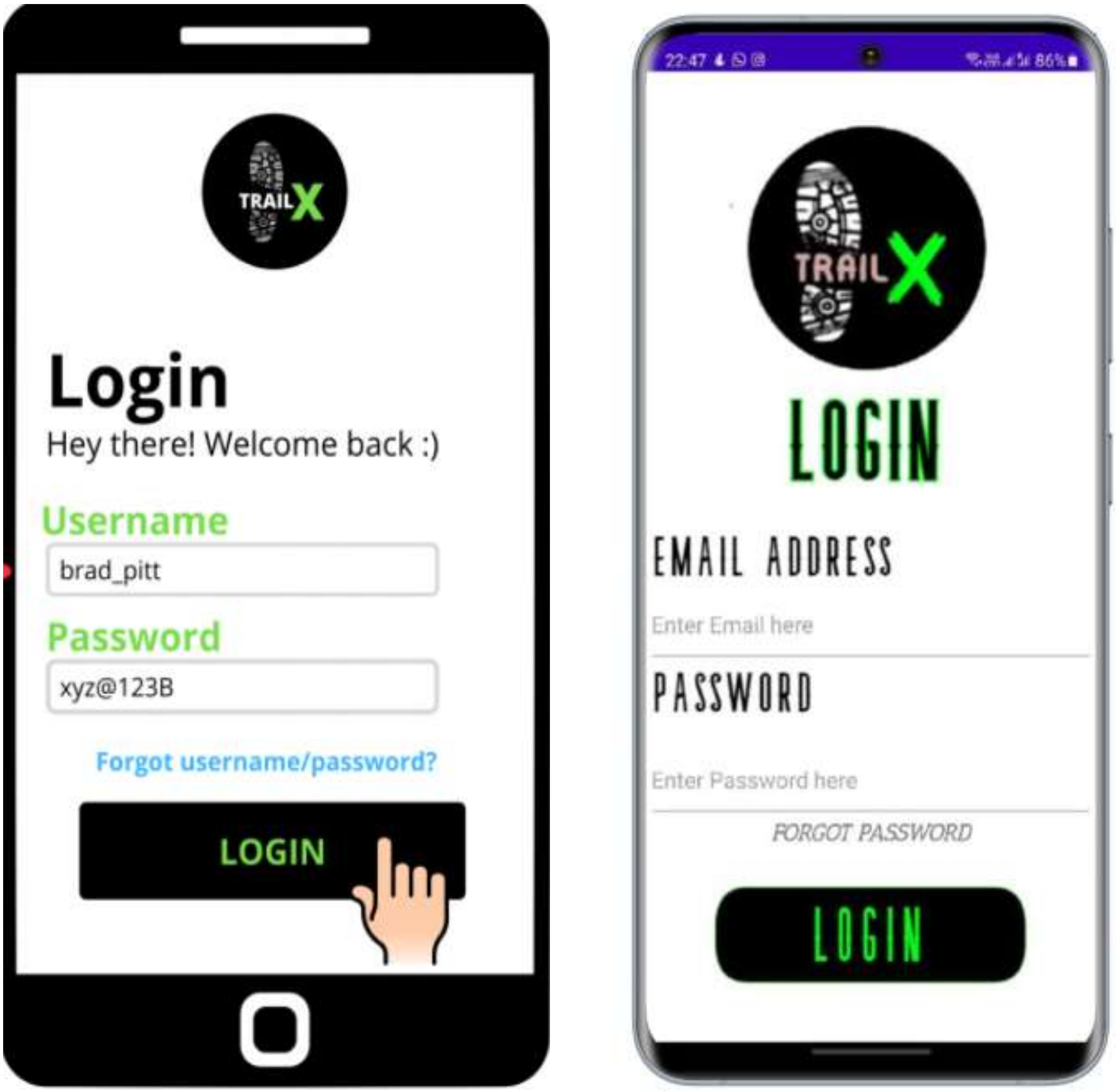
○ Female

○ Other

Screen 3: Login Screen

The Login Screen has simple input fields for Email Address and Password for registered users to enter. After the input fields are filled, clicking the “Login” button navigates the user to the Home Screen.

Figure 3: Login Screen: User Interface Mockup (left) versus final User Interface layout (right)



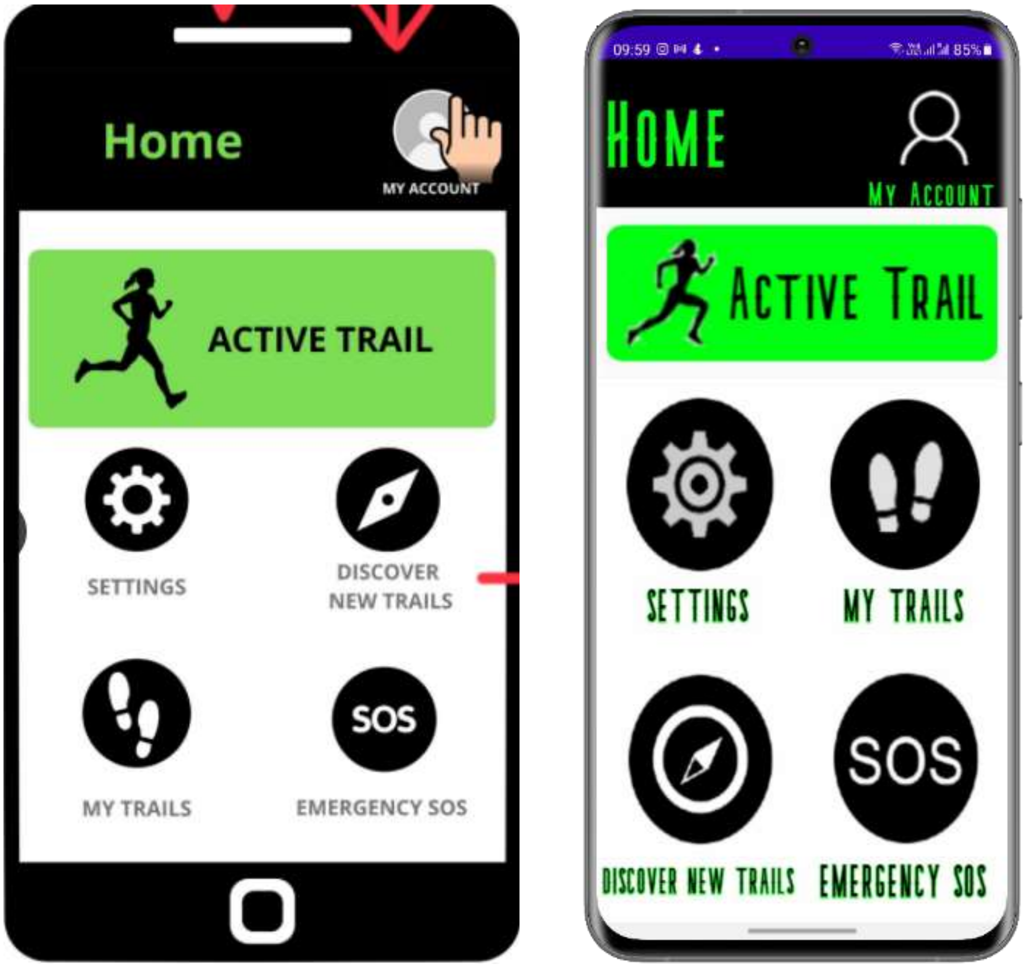
Login
Hey there! Welcome back :)
Username
brad_pitt
Password
xyz@123B

Screen 4: Home Screen

The Home Screen consists of all the functions that the application has to offer. It consists of commonly used icons on buttons which are also named for users' understandability. The redirecting buttons available on the home page are –

- "My Account" that redirects to Screen 5: My Account Screen, which displays user's saved information.
- "Active Trail" that redirects to Screen 6: Active Trails Screen, which displays the active trail.
- "Settings" that redirects to Screen 10: Settings Screen, which allows users to edit their saved information.
- "Discover New Trails" that redirects to Screen 8: Discover New Trails Screen, where the user can select a new trail.
- "Emergency SOS" that redirects to Screen 9: Emergency SOS Screen, which calls emergency services.

Figure 4: Home Screen: User Interface Mockup (left) versus final User Interface layout (right)



Screen 5: My Account Screen

The My Account Screen displays the user's information on Full Name, Age, Gender, Email, Height (in centimeters) and Weight (in kilograms).

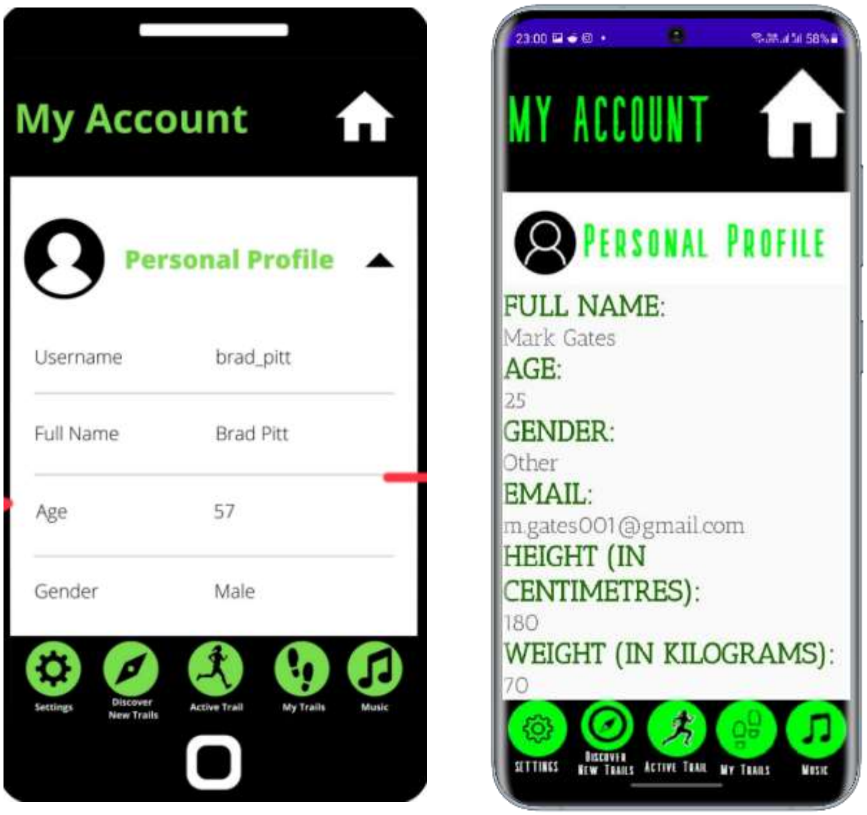
The icon bar at the bottom of the screen allows users to navigate to other screens easily. The icon bar is available consistently across all the screens excluding the Login Screen and Register Screen as it is the portal for users to enter the application and the Home Screen which consists of all the buttons. The icon bar is available to users as a shortcut to get to the commonly used functions rather than navigating through the Home Screen.

The buttons available on the icon bar are:

- "Settings" that redirects to Screen 10 - Settings Screen.
- "Discover New Trails" redirects to Screen 8 - Discover New Trails Screen.
- "Active Trail" that redirects to Screen 6 - Active Trails Screen.
- "My Trails" redirects to Screen 7 - My Trails Screen.
- "Music" that redirects to Screen 11 - Music Screen.

My Account Screen also consists of a home icon button that redirects the user to Screen 4: Home Screen.

Figure 5: My Account Screen: User Interface Mockup (left) versus final User Interface layout (right)



My Account

Personal Profile

Username	brad_pitt
Full Name	Brad Pitt
Age	57
Gender	Male

MY ACCOUNT

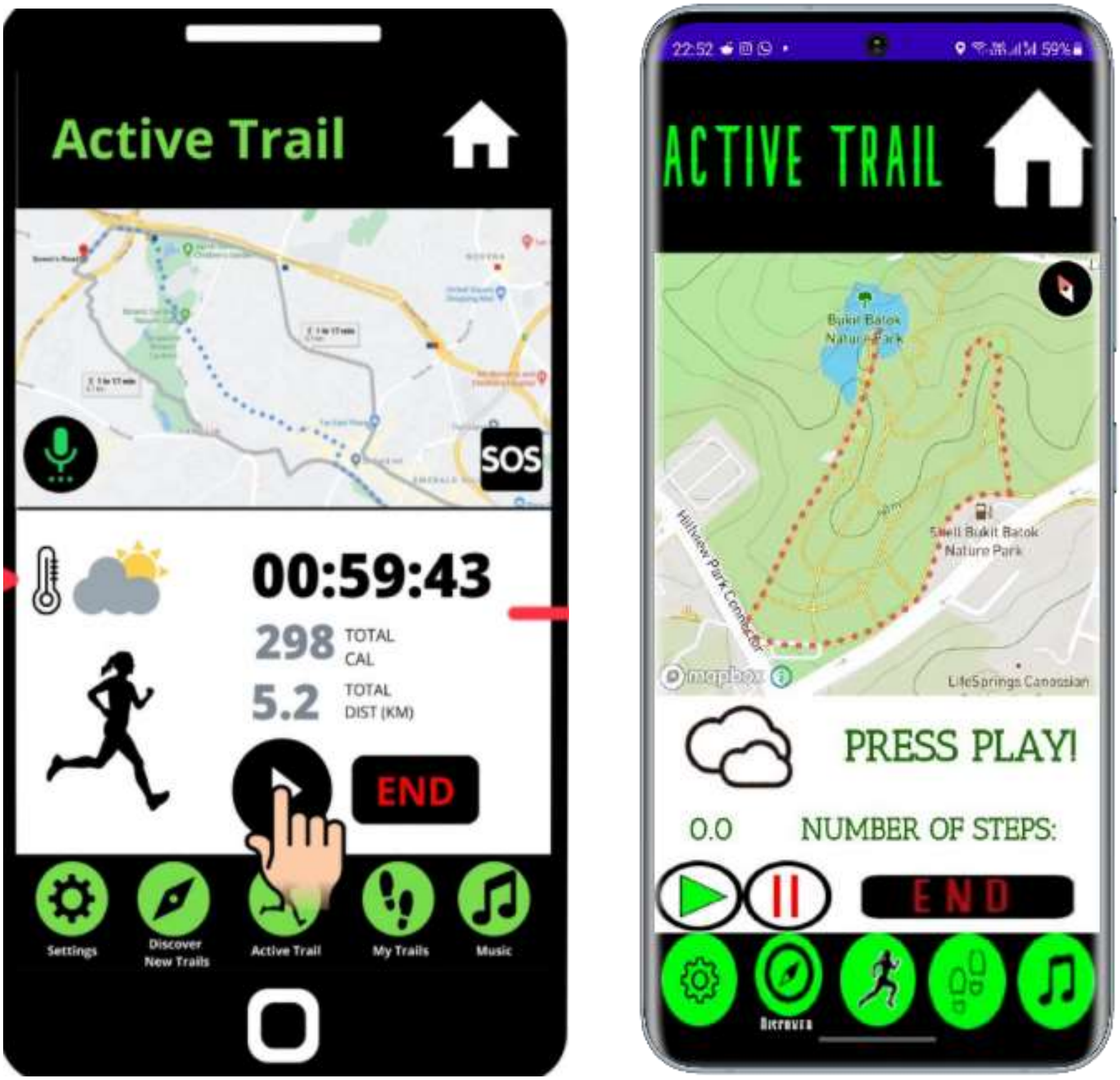
PERSONAL PROFILE

FULL NAME:
Mark Gates
AGE:
25
GENDER:
Other
EMAIL:
m.gates001@gmail.com
HEIGHT (IN CENTIMETRES):
180
WEIGHT (IN KILOGRAMS):
70

Screen 6: Active Trail Screen

The Active Trail Screen displays the map of the current trail and the location of the user. The screen also displays the step count and calories burnt by the user. A constant tab of the weather is available. The buttons for play, pause and end provide the users full control of when to start, pause and stop the trail.

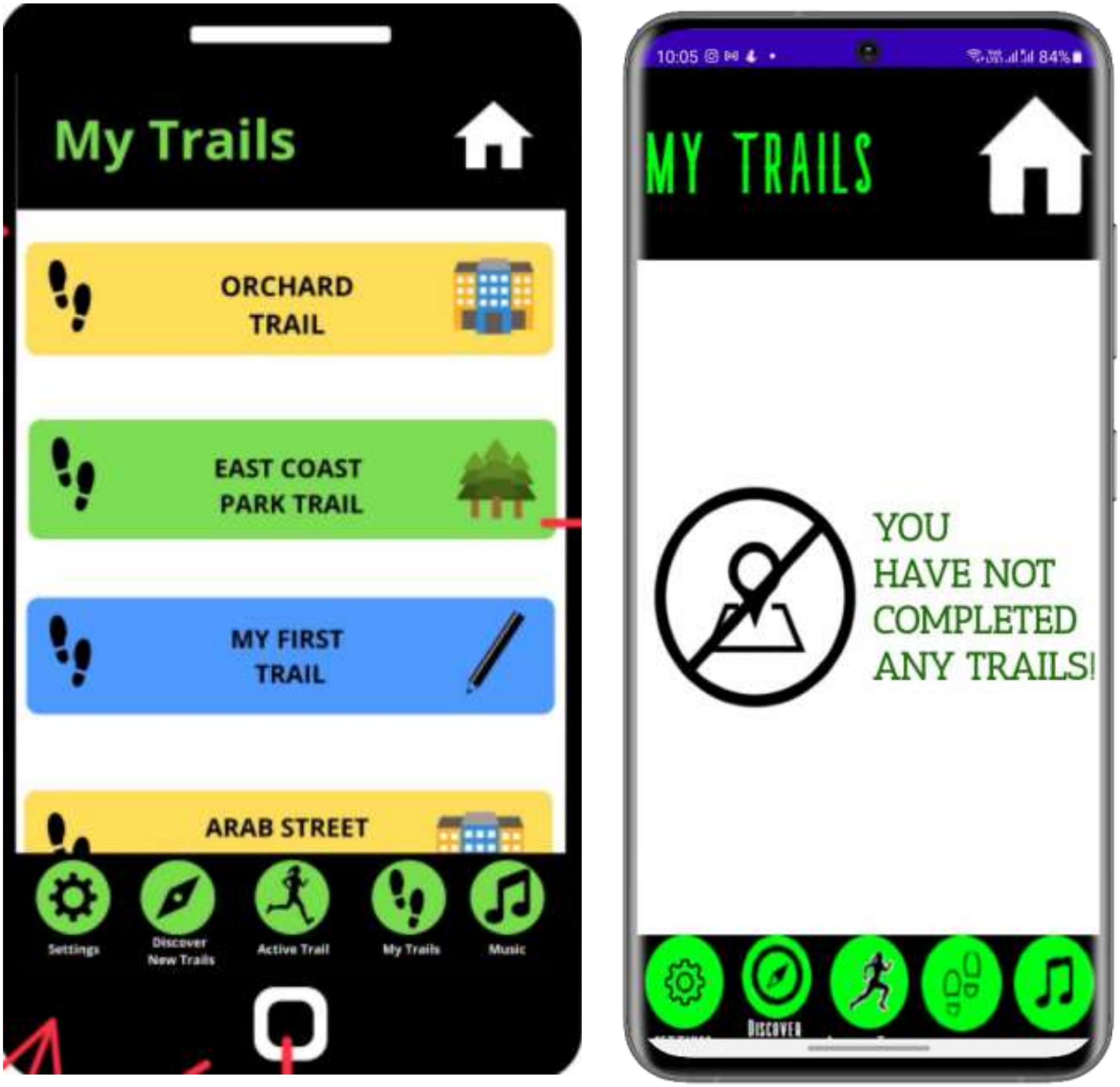
Figure 6: Active Trail Screen: User Interface Mockup (left) versus final User Interface layout (right)



Screen 7: My Trails Screen

My Trails Screen displays all the trails explored by the user.

Figure 7: My Trails Screen: User Interface Mockup (left) versus final User Interface layout (right)



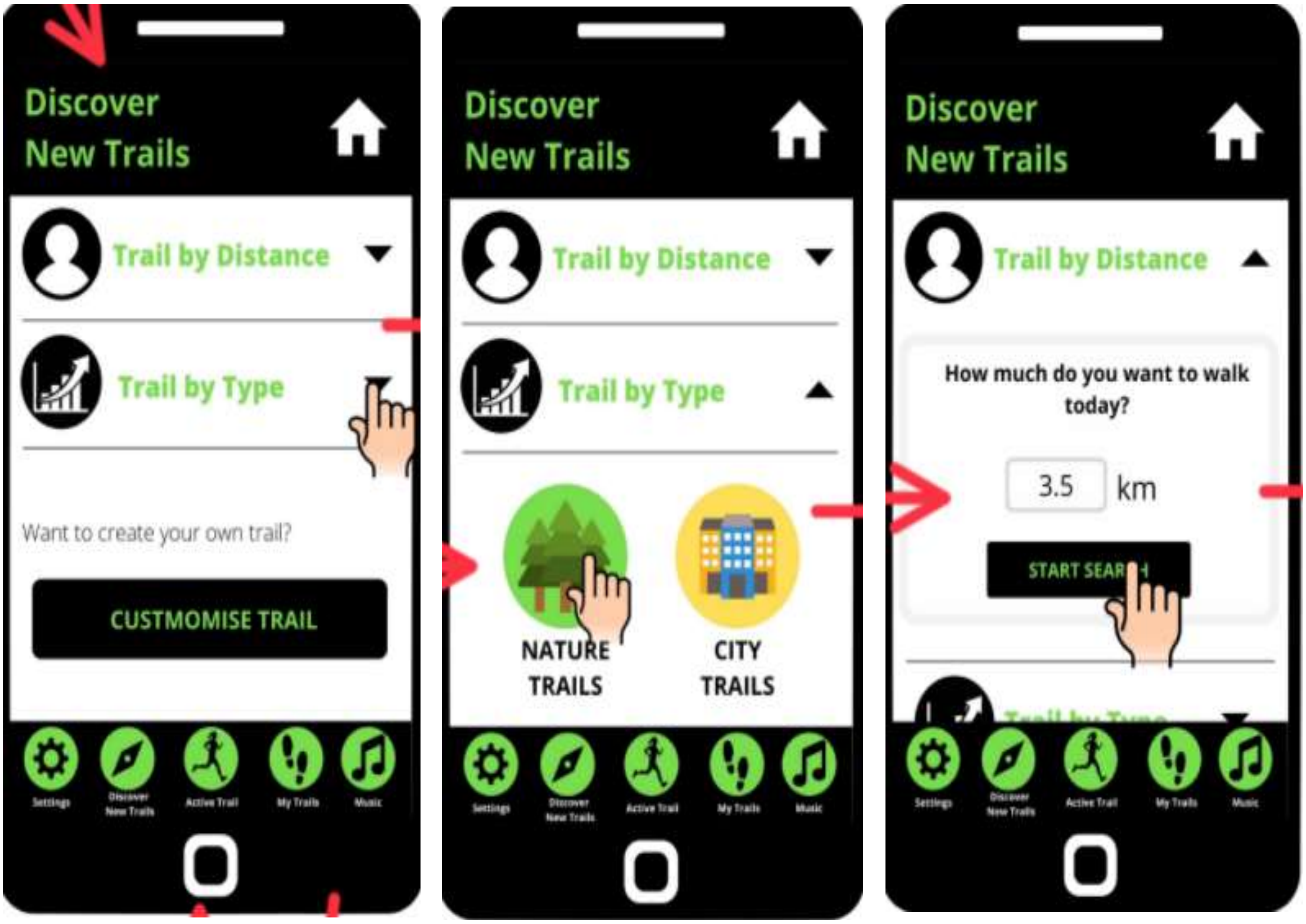
Screen 8: Discover New Trails Screen

The Discover New Trails Screen contains three sub-screens. Sub - Screen 1 of Screen 8 contains two buttons: “Trails by Distance” and “Trails by Type”.

The button “Trails by Distance” navigates the user to Sub - Screen 2 which contains an input field for entering distance in kilometers. According to the distance, Trail Name and Total distance will be displayed. After which clicking the button “Start Trail” redirects the user to Active Trails Screen.

The button “Trails by Type” navigates the user to Sub -Screen 3 which contains two buttons: “City Trails” and “Nature Trails”. Clicking on any of the two buttons displays the Trail Name and Trail Type. After which clicking the button “Start Trail” redirects the user to Active Trails Screen.

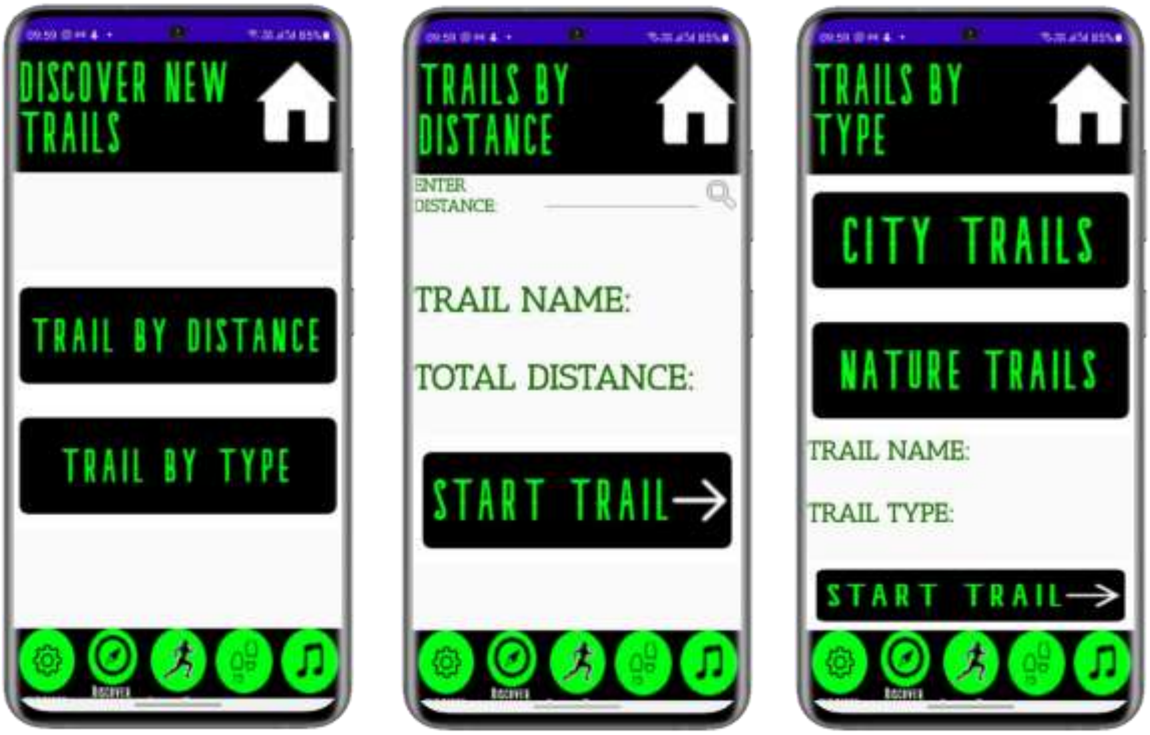
Figure 8a: Discover New Trails Screen: User Interface Mockup



Want to create your own trail?

How much do you want to walk today?

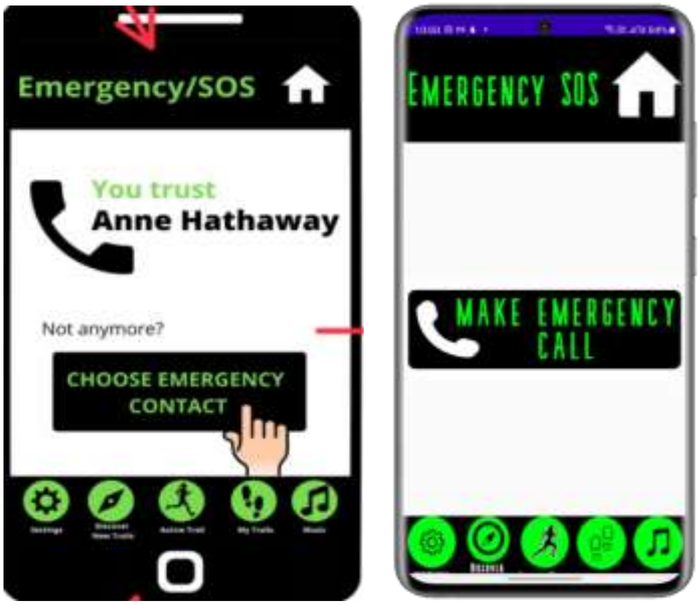
Figure 8b: Discover New Trails Screen: Final User Interface layout



Screen 9: Emergency SOS Screen

The Emergency SOS Screen contains a button "Make Emergency Call" which makes a call to emergency services. For testing purposes, the number used in code is **not** the real emergency hotline.

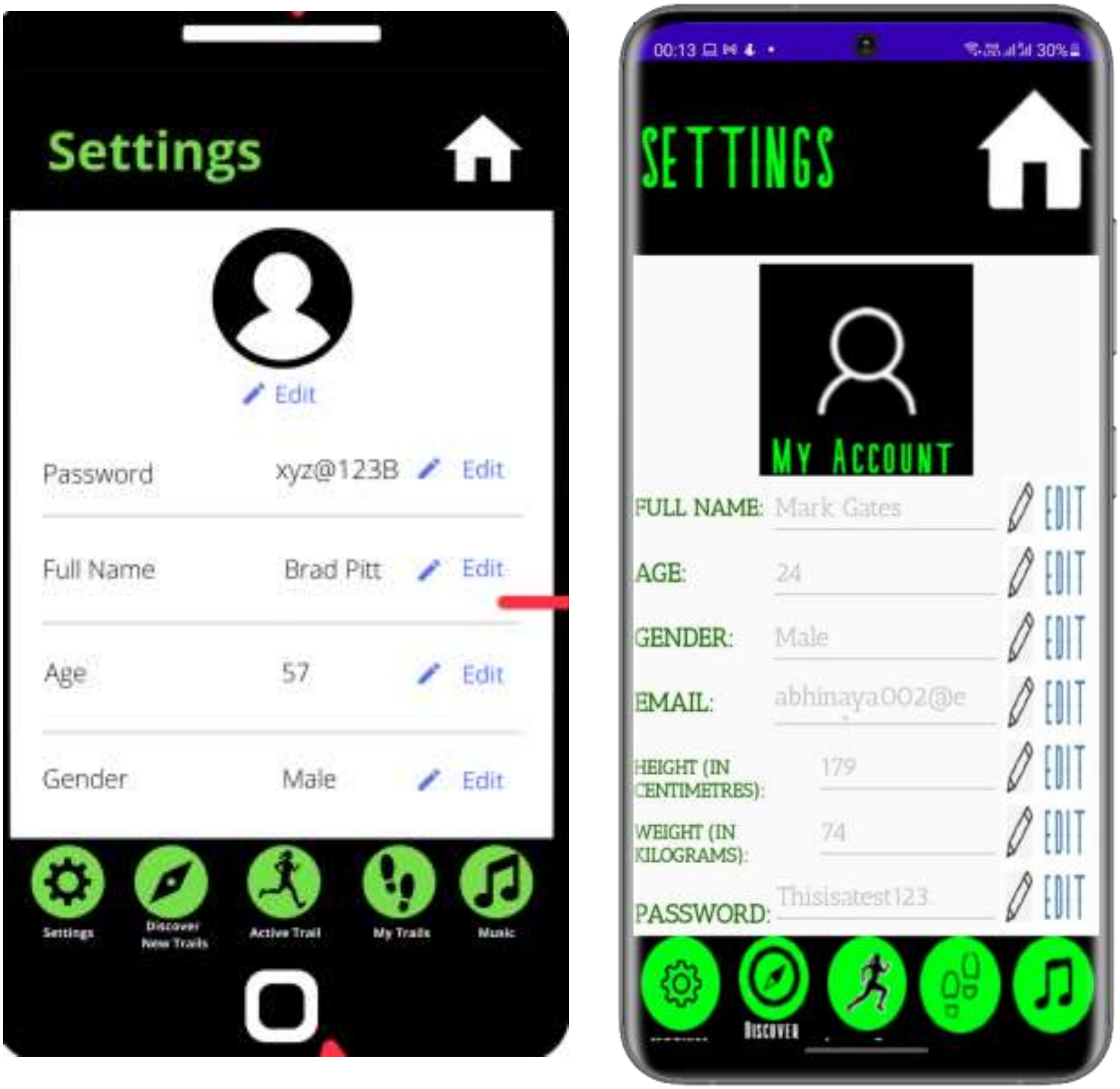
Figure 9: Emergency SOS Screen: User Interface Mockup (left) versus final User Interface layout



Screen 10: Settings Screen

The Settings Screen contains the user's information such as full name, age, gender, email, height (in centimeters), weight (in kilograms), username and password. The "Edit" button enables the user to change the aforementioned information.

Figure 10: Settings Screen: User Interface Mockup (left) versus final User Interface layout (right)



Settings

Edit

Password xyz@123B Edit

Full Name Brad Pitt Edit

Age 57 Edit

Gender Male Edit

SETTINGS

MY ACCOUNT

FULL NAME: Mark Gates EDIT

AGE: 24 EDIT

GENDER: Male EDIT

EMAIL: abhinaya002@e EDIT

HEIGHT (IN CENTIMETRES): 179 EDIT

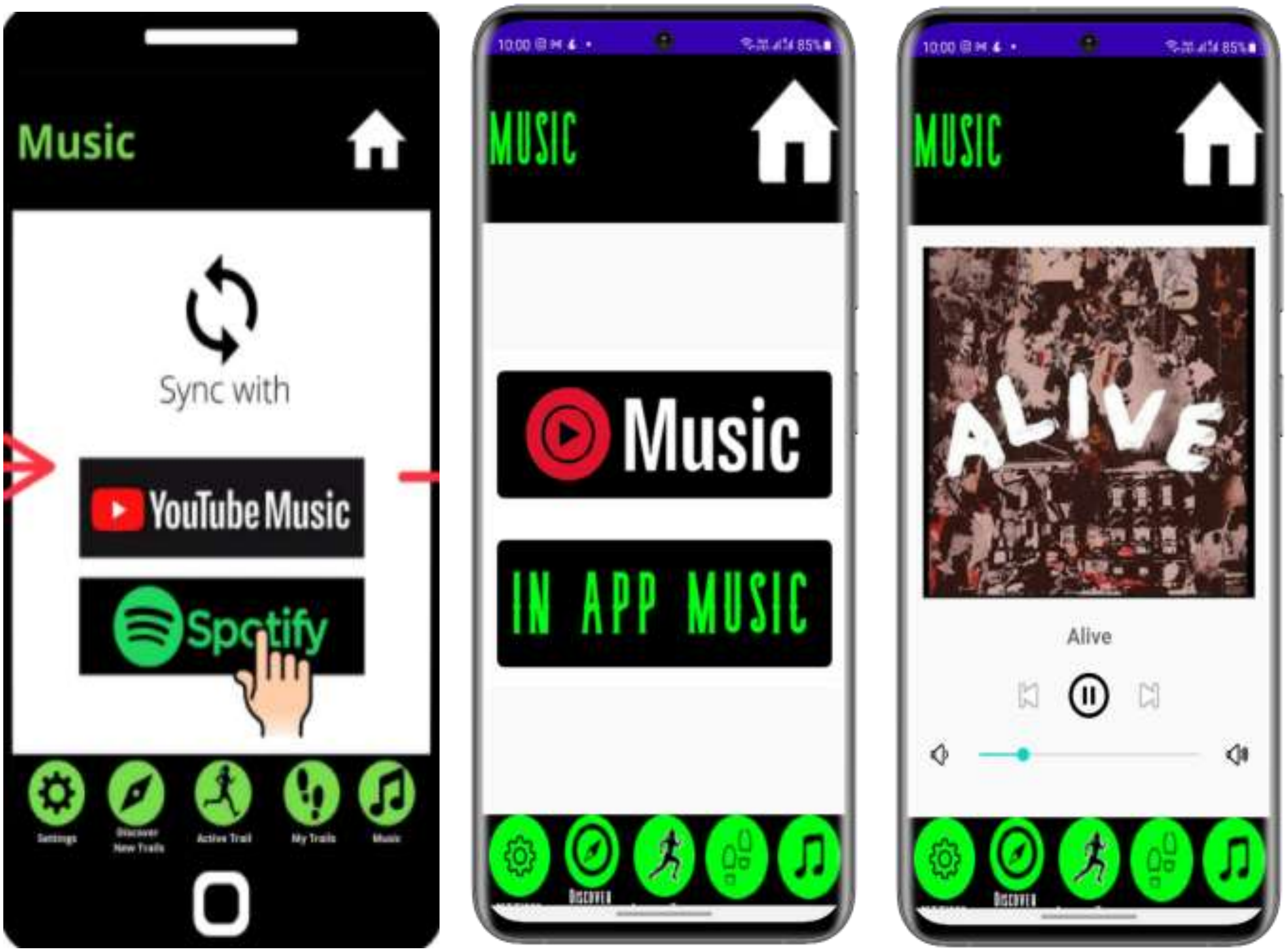
WEIGHT (IN KILOGRAMS): 74 EDIT

PASSWORD: Thisisatest123 EDIT

Screen 11: Music Screen

The Music Screen contains two buttons: “YouTube Music” and “In App Music”. YouTube Music button redirects the user to YouTube whereas the “In App Music” button redirects the user to the music player. The music player contains buttons such as the play, pause, rewind, and forward buttons. These buttons are used to play or pause the current song, rewind to the previous song and forward to the next song. The screen also contains a sliding button to increase or decrease the volume of the song.

Figure 11: Music Screen: User Interface Mockup (left) versus final User Interface layout (right)



3.2 Hardware Interfaces

The hardware aspect of the *TrailX* application is heavily dependent on the touch screen interface of a smartphone. The touch actions determine the next state of the application. Touch actions like swiping, tapping, scrolling, pinching and reverse pinching are to be used throughout the application. The application requires GPS modules, gyroscope sensors and Wi-Fi modules.

3.3 Software Interfaces

TrailX utilizes Firebase as a software for storing the user data in the form of a database. Firebase Android BoM version 26.8.0 is utilized for the same. Google services 4.3.5 is used to facilitate the communication between the application and the Google cloud server. The application uses Mapbox to display the trail route as well as the user location during an active trail. Mapbox version 8.5.1 is used for the purpose stated above. OpenWeather API version 2.0 is used to provide the user with live weather data during the active trail.

3.4 Communications Interfaces

The transfer of data between the application and Firebase takes place using WebSockets to keep the communications real-time. Hyper Text Transfer Protocol (HTTP) is utilized to communicate with the OpenWeather API. Secure Sockets Layer (SSL) / Transport Layer Security (TLS) is used for the communication between Mapbox API and the application.

4. System Features

4.1 Start Screen

4.1.1 Description and Priority

The <u>Start Screen</u> allows the user to access the features of the application by either logging in to an existing user account or registering as a new user by providing the necessary details. This screen has a High priority as it is important for the user to be successfully logged in to access any feature of the application.

4.1.2 Stimulus/Response Sequences

The application must be downloaded by the user onto a compatible Android smartphone. The Start Screen is displayed when the application is opened.

4.1.3 Functional Requirements

1. Upon start, users must press either the "Register" or the "Login" button

1.1. No/Null Response from the user will not change the current state of the mobile application.

4.2 Registration Screen

4.2.1 Description and Priority

The Registration Screen allows the users to create a profile for themselves and create an account with the application using their email address. This screen has a High priority as user accounts are vital for accessing the application's features.

4.2.2 Stimulus/Response Sequences

The Register button must be pressed on the Start Screen for the user

4.2.3 Functional Requirements

2. The user is directed to the Registration Screen, the user must input their full name, age, gender, email address, height, weight, username, and password.

2.1. Full Name must be of string(text) data type.

2.1.1. If the Full Name input field is left blank/unanswered, the user is prompted to fill up the field.

2.1.2. If the input in the Full Name field is not (string(text)) data type, the user is prompted to refill the field with the correct data type (string(text)) value.

2.2. Age must be of integer data type.

2.2.1. If the age input field is left blank/unanswered, the user is prompted to fill up the field.

2.2.2. If the input value for age is not valid, the user is prompted to refill the field with the correct data type value.

2.3. Gender must be chosen from a radio button group with the following options - Male, Female and Others.

2.3.1. If the Gender radio button is left unpicked, the user is prompted to pick an option and fill up the field.

2.4. Email must be of string data type.

2.4.1. User must input a valid email address.

2.4.1.1. A valid email address must incorporate '@'

2.4.1.2. A valid email address must not end with a text type character or string.

2.4.2. If the email input field is left blank/unanswered, the user is prompted to fill up the field.

2.4.3. If the input in the Email field is not text/string data type, the user is prompted to refill the field with the correct data type (string) value.

2.5. Height must be of integer data type.

2.5.1. User must input a valid value for height.

2.5.1.1. A valid value of height is represented in centimeters.

2.5.1.2. A valid value of height is bounded between 50 centimeters and 300 centimeters, inclusive of 50 and 300.

2.5.2. If the Height field is left blank/unanswered, the user is prompted to fill up this field.

2.6. Weight must be of integer data type.

2.6.1. User must input a valid value for weight.

2.6.1.1. A valid value for weight is represented in kilograms.

2.6.1.2. A valid value for weight is bounded between 20 kilograms and 500 kilograms, inclusive of 20 and 500.

2.6.2. If the Weight field is left blank/unanswered, the user is prompted to fill up this field.

2.7. Username must be of string data type.

2.7.1. User must input a valid username.

2.7.1.1. A valid username must be minimum 6 characters.

2.7.1.2. A valid username cannot contain characters other than text and integer datatype and the two special characters (. , @).

2.7.2. If the username field is left blank/unanswered, the user is prompted to fill up this field.

2.7.3. If the input value of username does not meet requirements 2.7.1.1 and 2.7.1.2, the user is prompted to refill this field with the correct requirements.

2.8. Password must be of string data type.

2.8.1. User must input a valid password.

2.8.1.1. A valid password must be minimum 8 characters.

2.8.1.2. A valid password must contain at least 1 capital letter (A-Z)

2.8.1.3. A valid password must contain at least 1 number (0-9)

2.8.1.4. A valid password must contain at least 1 of the following characters - ('.' , '_', '-')

2.8.2. If the password field is left blank/unanswered, the user is prompted to fill up this field.

2.8.3. If the input value of password does not meet the requirements of 2.8.1.1, 2.8.1.2, 2.8.1.3, 2.8.1.4, the user is prompted to refill this field with the correct requirements.

2.9. The user must press the "Register" button to proceed with the registration.

2.9.1. If the "Register" button does not receive input from the user, the state of the application does not change.

2.9.2. If even 1 of the 8 requirements are not fulfilled (2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8) the user is not permitted to proceed.

2.9.3. If all the requirements stated in 2.9.2 are satisfied, the user is redirected to the Home Screen.

4.3 Login Screen

4.3.1 Description and Priority

The Login Screen allows the users to access the application by using a pre – created user account with the *TrailX* Application. This screen has a High priority as User accounts are vital for accessing the application's features.

4.3.2 Stimulus/Response Sequences

The Login button must be pressed on the Start Screen for the user

4.3.3 Functional Requirements

3. When the user is redirected to the Login Screen, the user is prompted to input their email and password

3.1. The user must enter a valid email.

3.1.1. A valid email is one that was input by the user in the Registration Screen (Functional Requirement 2.4)

3.1.2. A valid email must exactly match the one that was registered by the user in the Registration Screen.

3.1.2.1. If the input email does not match the email that was used to register, the user is prompted to refill the username field.

3.1.3. If the email field is left blank/unanswered, the user is prompted to fill up the field.

3.2. The user must enter a valid password.

3.2.1. A valid password is one that was input by the user in the Registration Screen (Functional Requirement 2.8)

3.2.2. A valid password must exactly match the one that was registered by the user in the Registration Screen.

3.2.2.1. If the input password does not match the password that was used to register, the user is prompted to refill the password field.

3.2.2.2. If the user forgets the password, the user must click on the "Forgot username/password" button to retrieve login information. (More in Functional Requirement 3.3)

3.2.3. If the password field is left blank/unanswered, the user is prompted to fill up this field.

3.3. When password is forgotten, the user must press the 'Forgot password' button.

3.3.1. In the email field the user must input a valid email address.

3.3.1.1. A valid email address is one that matches exactly as the one input during registration.

3.3.2. If the email address is invalid, the user is prompted to refill the field with the valid address.

3.3.3. If the email field is left blank/unanswered, the user is prompted to fill up the field.

3.3.4. If the requirements 3.3.1, 3.3.2 and 3.3.3 are satisfied, the system sends an email to the input email address, containing a password reset link.

3.3.5. The user must go back to the login page and fill in the contents of the email in the username and password fields.

3.4. The user must press the "Login" button to proceed with the login.

3.4.1. If the "Login" button does not receive input from the user, the state of the application does not change.

3.4.2. If even 1 of the 2 requirements are not fulfilled (3.1, 3.2) the user is not permitted to proceed.

3.4.3. If all the requirements stated in 3.4.2 are satisfied, the user is redirected to the Home Screen.

4.4 Home Screen

4.4.1 Description and Priority

The Home Screen allows the user to access the various aspects of the application, namely, Active Trails Screen, Discover New Trails Screen, Music Screen and My Trails Screen. This Screen has a high importance since it acts as the main interface for other features in the application.

4.4.2 Stimulus/Response Sequences

After a successful Login or Registration, the user is redirected to the Home Screen of the application.

4.4.3 Functional Requirements

4. The Home Screen has the following components

4.1. When the "My Account" button is clicked, the user is redirected to the My Account Screen. (Section 6.0)

4.2. When the Settings button is clicked, the user is redirected to the Settings Screen (Section 7.0)

4.3. When the Discover New Trails button is clicked, the user is redirected to the Discover New Trails Screen (Section 8.0)

4.4. When the My Trails button is clicked, the user is redirected to the My Trails Screen (Section 9.0)

4.5 When the Active Trail button is clicked, the user is redirected to the Active Trails Screen. (Section 10.0)

4.5 Emergency SOS Screen

4.5.1 Description and Priority

The Emergency SOS Screen allows the user to place and Emergency SOS Call. This feature is of High priority to ensure the safety of our users during the walking trails.

4.5.2 Stimulus/Response Sequences

The user is already logged on to the system and has also proved access to the phone's stored contacts. The user is navigated to this screen on clicking the Emergency SOS button on the Home Screen.

4.5.3 Functional Requirements

5. When the user is redirected to the Emergency SOS Screen, the user can place an emergency call to call for help.

5.1. The user must select the "Emergency/SOS" button from the home page.

5.2. When the "Emergency/SOS" button is clicked by the user, a call is placed to help services from the user's phone number.

4.6 My Account Screen

4.6.1 Description and Priority

On the My Account Screen, the user can view all the information stored related to the User's profile on the database. This screen forms a Medium priority as it only displays user profile information. This information can be edited in the Settings Screen.

4.6.2 Stimulus/Response Sequences

The user is already logged on to the system and has also proved access to the internet. The user is navigated to this screen on clicking the My Account button on the Home Screen.

4.6.3 Functional Requirements

6. When the user is redirected to the My Account Screen, the user can view Personal Profile.

6.1. The details displayed are the username, password, full name, age, gender, email address, height, and the weight of the user.

6.1.1. The height of the user is displayed in cm.

6.1.2. The weight of the user is displayed in kgs.

4.7 Settings Screen

4.7.1 Description and Priority

The Settings Screen allows the user to edit their personal information. This screen is of High importance as it is important to allow the user to keep their information up to date.

4.7.2 Stimulus/Response Sequences

The user is already logged on to the system and has also proved access to the internet. The user is navigated to this screen on clicking the Settings button on the Home Screen.

4.7.3 Functional Requirements

7. When the user is redirected to the Settings Screen, the user can edit their personal profile.

    7.1. The user can edit the height and weight data stored in the profile.

        7.1.1. The user must input a valid height.

            7.1.1.1. A valid value of height is represented in centimeters.

            7.1.1.2. A valid value of height is bounded between 50 centimeters and 300 centimeters, inclusive of 50 and 300.

        7.1.2. If the Height field is left blank/unanswered, the user is prompted to fill up this field.

        7.1.3. If the input value of height is not an integer, the user is prompted to refill the field with the correct data type (integer)

        7.1.4. The user must input a valid weight.

            7.1.4.1. A valid value for weight is represented in kilograms.

            7.1.4.2. A valid value for weight is bounded between 20 kilograms and 500 kilograms, inclusive of 20 and 500.

        7.1.6. If the Weight field is left blank/unanswered, the user is prompted to fill up this field.

        7.1.7. If the input value of weight is not an integer, the user is prompted to refill the field with the correct data type (integer)

7.2. The user can change the gender by editing the Gender field.

7.2.1. If the Gender field is left empty, the user is prompted to pick an option and fill up the field.

7.3. The user can change the password by editing the password field.

7.3.1. The new password entered must be valid.

7.3.1.1. A valid password must be minimum 8 characters.

7.3.1.2. A valid password must contain at least 1 capital letter (A-Z)

7.3.1.3. A valid password must contain at least 1 number (0-9)

7.3.1.4. A valid password must contain at least 1 of the following characters - ('.' , '_', '-')

4.8 Discover New Trails Screen

4.8.1 Description and Priority

The Discover New Trails Screen allows the user to discover all the trails currently available in the system by searching either using "Trails by Type" or "Trails by Distance". This screen is of High importance as it allows the user to explore new trails for them to take.

4.8.2 Stimulus/Response Sequences

The user is already logged on to the system and has also proved access to the internet. The user is navigated to this screen on clicking the "Discover New Trails" button on the Home Screen.

4.8.3 Functional Requirements

8. When the user is redirected to the Discover New Trails screen, the user can view 2 buttons: "Trail by distance" and "Trail by type".

8.1. The "Trail by distance" button redirects the user to a new screen which prompts the user to enter a distance. The total distance entered is in kilometers. On entering the distance, a list of available trails with distance approximately equal (with a difference of not more than 1 kilometer) to the user's input is displayed.

8.1.1. The user must enter a valid numeric distance, between 0 to 50.

8.1.1.1. If the input is invalid, the user is prompted to re-enter a numeric value.

8.2. The "Trail by type" button redirects the user to a new screen that contains 2 buttons: "Nature Trail" and "City Trail". The user is prompted to select one of these two options.

8.2.1. If the user selects "Nature trail", the name of a randomly selected nature trail is displayed.

8.2.2. If the user selects "City trail", the name of a randomly selected city trail is displayed.

8.2.3. The user can start one of these trails by clicking "Start Trail" which redirects the user to the Active Trails Screen.

4.9 My Trails Screen

4.9.1 Description and Priority

The My Trails Screen allows the user to access the trails that have been saved by the user for future purposes. This screen is of Medium importance as it allows the user to view their trail history.

4.9.2 Stimulus/Response Sequences

The user is already logged on to the system and has also proved access to the internet. The user is navigated to this screen on clicking the "My Trails" button on the Home Screen.

4.9.3 Functional Requirements

9. When the user is directed to the My Trails Screen, the following information is presented: name of the trail and snapshot of the trail route.

9.1. If the user has not completed any trail, this section states that the user has not completed any trail.

4.10 Active Trail Screen

4.10.1 Description and Priority

The Active Trails Screen provides all the relevant information that is pertinent to the currently active walking trail. Since this Page forms the core of the *TrailX* application, it has the highest priority.

4.10.2 Stimulus/Response Sequences

The user is already logged on to the system and has also proved access to the internet. The user is navigated to this screen on clicking the "Active Trail" button on the Home Screen.

4.10.3 Functional Requirements

10. When the user is redirected to the Active Trails Screen, the user can view the following sections:

10.1. The route map of the trail

10.1.1. The user can interact with the map for better understanding of the route as well as their location.

10.2. The user can either listen to preloaded songs on an in – built music player or play music from YouTube while on an active trail (furthered in Section 11.0)

10.2.1. The user must press the "Music" button at the bottom of the screen to access the available music streaming options.

10.3. The user can get live information on weather at their location on the day of their active trail.

10.3.1 The weather information is displayed in the form of an image.

10.4. The trail information section gives the user the following information on current session.

10.4.1. Step count section since the beginning of the trail.

10.4.2. Time elapsed in HH:MM:SS format

10.4.3. Calories burned in kcal.

10.4.4. Pause, Play, and END controls for users to control the pace according to their comfort.

10.4.4.1. When user presses pause, the timer is paused.

10.4.4.2. Pressing play resumes the timer.

10.4.4.3. END button allows user to end the active trail and saves the name and snapshot of the trail to the history.

10.4.4.3.1. This trail must then be found in the future from the My trails Screen.

4.11 Music Screen

4.11.1 Description and Priority

The Music Screen allows the user to play music using an in-app music player or using an external agent - YouTube. This screen has a low priority as the music is present as an added feature to accompany the user on the trail.

4.11.2 Stimulus/Response Sequences

The user is already logged on to the system and has also proved access to the internet. The user is navigated to this screen on clicking the “Music” button on the Home Screen.

4.11.3 Functional Requirements

11. If a user wishes to play songs while on the trail, he or she will be able to play music in-app or on YouTube.

11.1. If the user chooses YouTube, he/she gets redirected to the YouTube Application where they can play music.

11.2. If the user chooses in-app music, he or she can listen to the music available in the *TrailX* app.

11.2.1. In the In-App Music Screen, the following functions are performed by the user.

11.2.1.1. When the user clicks the “Pause” button, the song pauses.

11.2.1.2. When the user clicks the "Play" button, the song starts playing.

11.2.1.3. When the user clicks the "Forward" button, the next song in the queue starts playing.

11.2.1.3.1. If the user is at the end of the queue, the song at the start of queue starts playing.

11.2.1.4. When the user clicks the "Rewind" button, the previous song in the queue starts playing.

11.2.1.4.1. If the user is at the beginning of the queue, the last song at the end of the queue starts playing.

11.3. If the user does not press any of the options or take any actions, the page in which the user is will remain still and wait for an input to be entered.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The application must acknowledge all user queries within 20 seconds.
- Querying of the Mapbox API must take less than 10 seconds to respond.
- The application must be able to support at – least 40 queries per minute.
- In case of failure of app, the platform must be available to the users again within 2 hours.
- The My Trails Screen must be updated within 5 minutes of completing or ending a trail.
- After a system reboot, the application must fully restore functionality within 2 minutes.

5.2 Safety Requirements

- The application utilizes the user's location to denote their current location on the map. No location information is stored in the application's system for safety purposes.
- The user's physical activity is accessed to display a record of the step count but is never stored.

5.3 Security Requirements

- The application shall not share the user's application usage history with third-party companies.
- The user's account information will not be shared with any third-party companies.

5.4 Software Quality Attributes

- The application will provide accurate and correct results for all user queries.
- The user will be provided with up-to-date trail information and correct distances.
- The application must be available to users for at – least 95% of the day.
- The mean time between system failures must be at – least one month.
- The buttons for the major features of the application on the home page will be made into icons for better usability and UX.
- The user must be able to learn how to use the various features of the application within the first 5 minutes of seeing the interface.
- The user should be able to reach their profile within 3 clicks.
- The mobile application must be compatible with all devices that are currently running Android 11.
- The Mapbox API must be accessible from within the application.

5.5 Business Rules

Our target users are Singaporeans and PRs with a valid NRIC, or foreigners with a valid FIN and aged 17 years or older (based on birth year) at point of registration.

Users can register with their personal information such as full name, age, gender, email, height, weight, and password. These details can be viewed by the user in the My Account Screen. Users are also allowed to edit this information in the Settings Screen. A user can only edit their information and cannot view that of other users registered to the system. Only the admin can manage the information of all the users in the database.

Users can choose the type of trail or the distance of a trail though they are restricted from selecting the trail. The trail based on the type selected by a user is allocated randomly which can be controlled by the admin.

Users can observe their location during an active trail only after the users decide to permit the application to access their location. They are deprived of changing the route, start location or end location of the trail. The privilege of handling the trail features is purely with the admin. Apart from these, users can also access the trails they have completed in My Trails Screen, along with the admin. Users have been given the privilege to make emergency calls to emergency services. Users have control in permitting the application to utilize the call features of the mobile phone. However, the admin has the authority to set the contact for emergency calls.

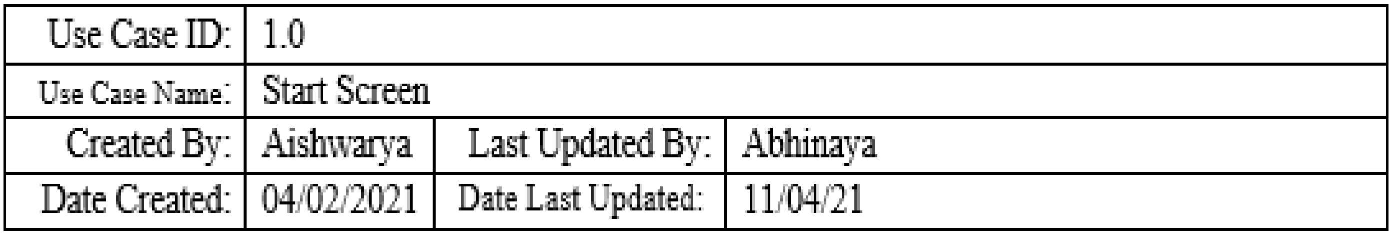
Users can constantly access the weather during the active trail but are not permitted to change the weather. The step count and calories burnt are displayed based on the user's movement and hence they are in control of these features. The way the steps and calories burnt are calculated is managed by the admin.

Users are authorized to access the songs available in the In-App Music Screen. They can perform functions such as play, pause, forward to the next song, rewind to the previous song and adjust the volume of the songs. They are restricted from adding or removing any songs from the in-app music playlist. The admin manually adds or removes songs from the in-app music.

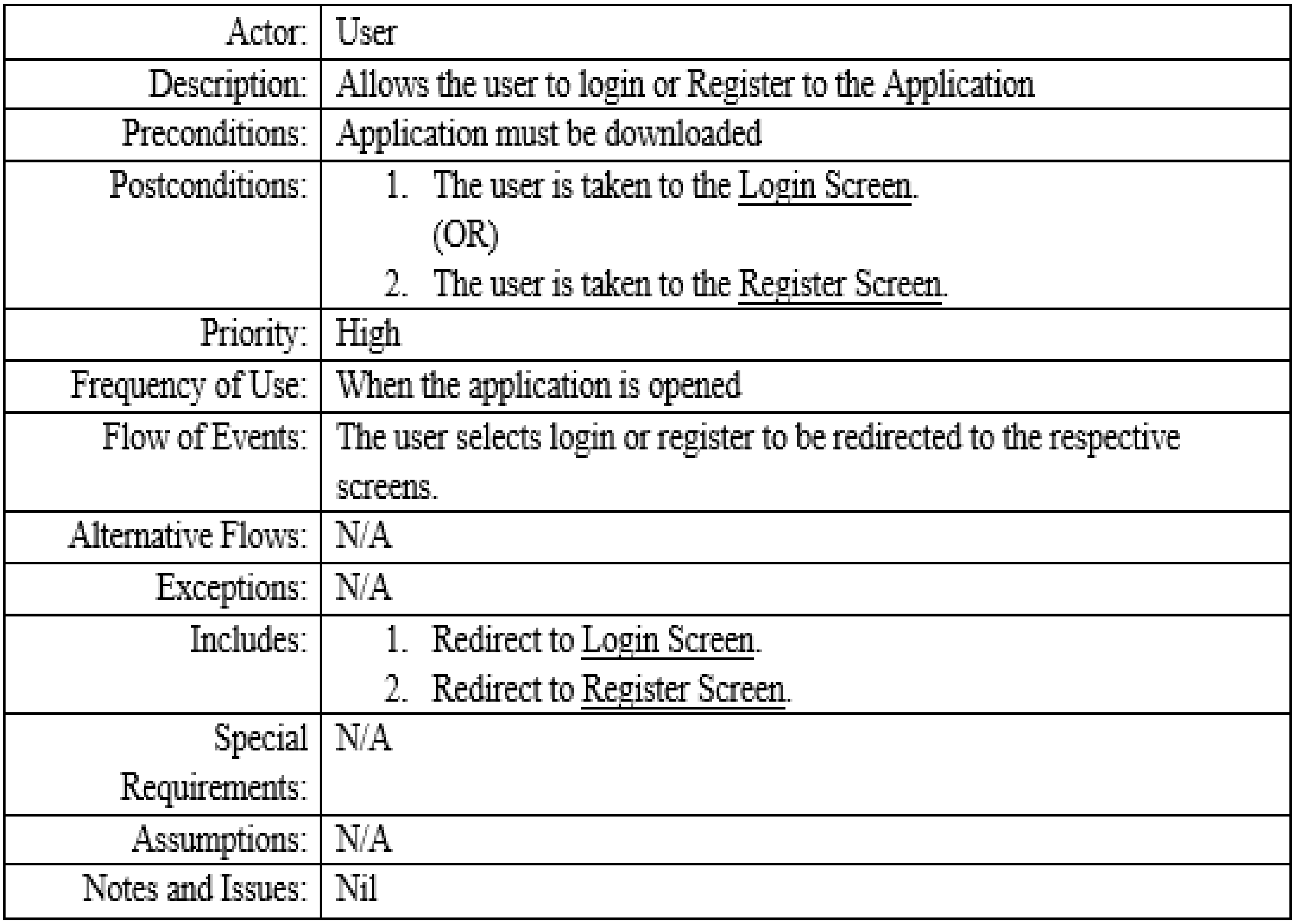
6. Use Case Descriptions

6.1 Start Screen

Table 1: Start Screen: Use Case Description



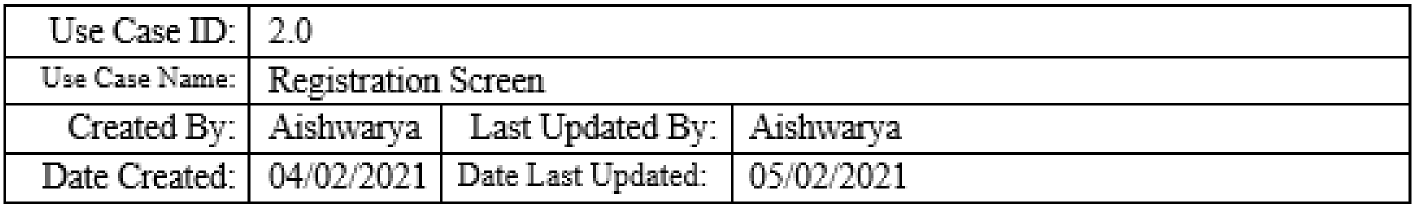
Use Case ID:	1.0		
Use Case Name:	Start Screen		
Created By:	Aishwarya	Last Updated By:	Abhinaya
Date Created:	04/02/2021	Date Last Updated:	11/04/21



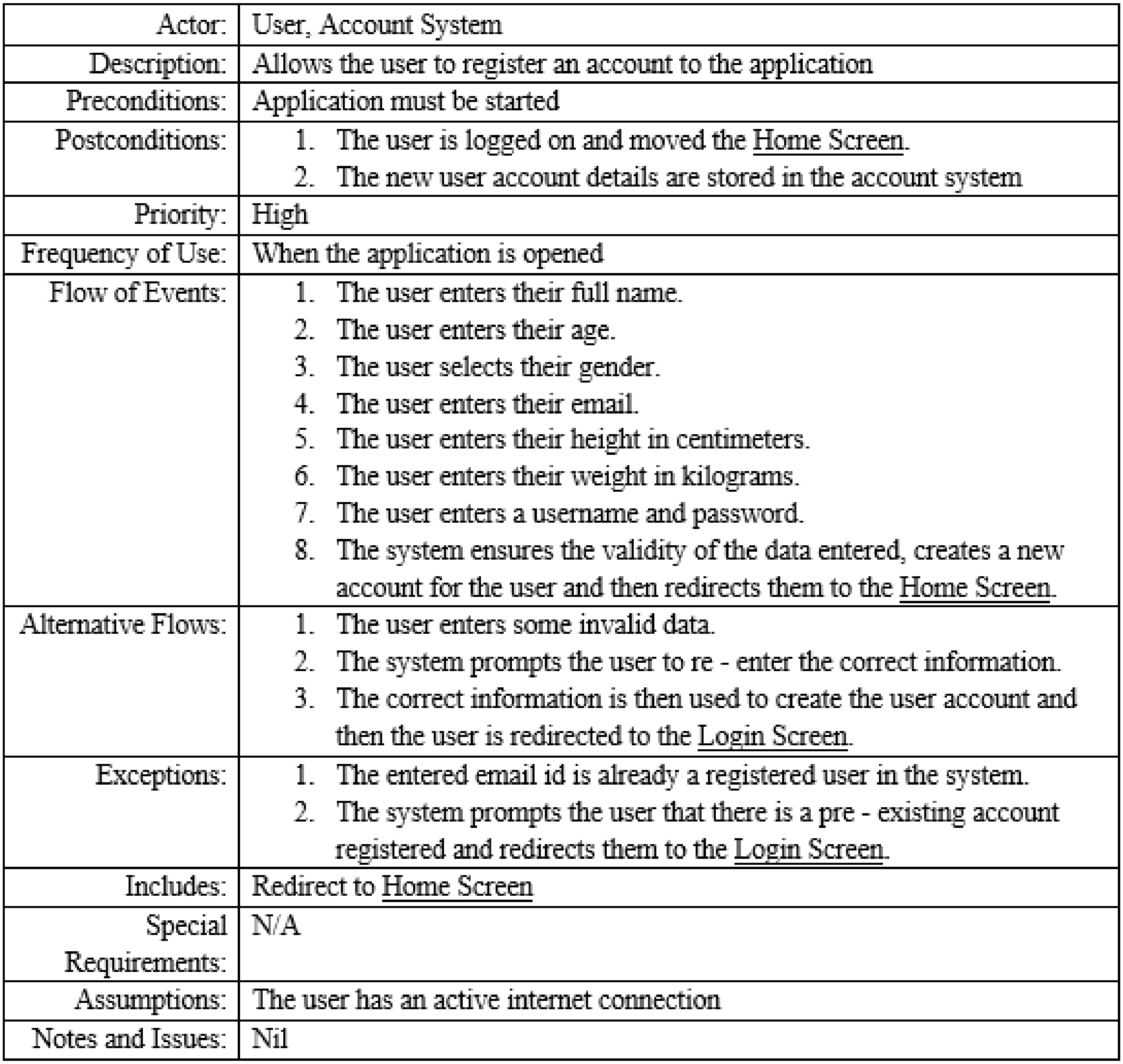
Actor:	User
Description:	Allows the user to login or Register to the Application
Preconditions:	Application must be downloaded
Postconditions:	1. The user is taken to the Login Screen. (OR) 2. The user is taken to the Register Screen.
Priority:	High
Frequency of Use:	When the application is opened
Flow of Events:	The user selects login or register to be redirected to the respective screens.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	1. Redirect to Login Screen. 2. Redirect to Register Screen.
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	Nil

6.2 Registration Screen

Table 2: Registration Screen: Use Case Description



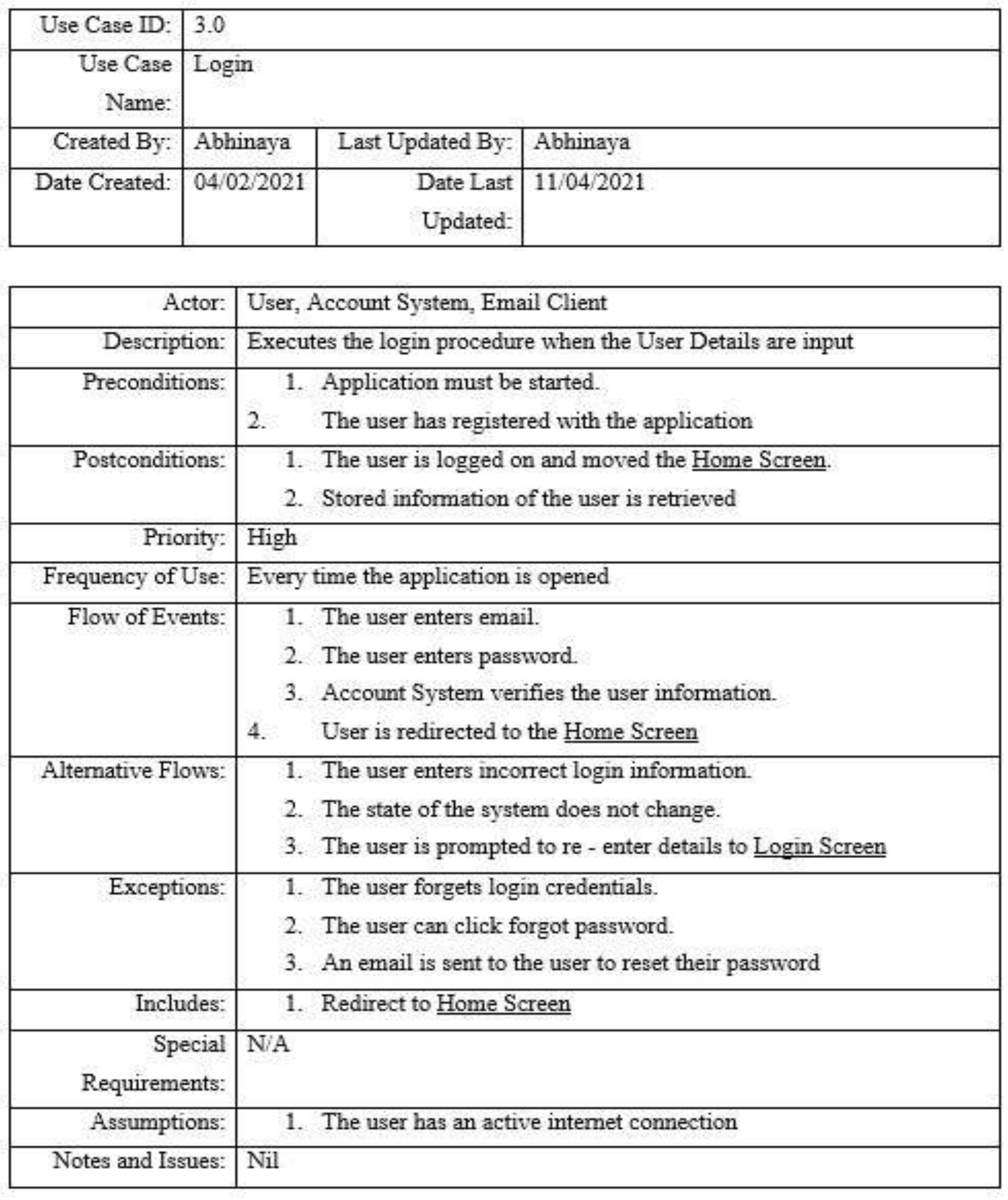
Use Case ID:	2.0		
Use Case Name:	Registration Screen		
Created By:	Aishwarya	Last Updated By:	Aishwarya
Date Created:	04/02/2021	Date Last Updated:	05/02/2021



Actor:	User, Account System
Description:	Allows the user to register an account to the application
Preconditions:	Application must be started
Postconditions:	1. The user is logged on and moved the Home Screen. 2. The new user account details are stored in the account system
Priority:	High
Frequency of Use:	When the application is opened
Flow of Events:	1. The user enters their full name. 2. The user enters their age. 3. The user selects their gender. 4. The user enters their email. 5. The user enters their height in centimeters. 6. The user enters their weight in kilograms. 7. The user enters a username and password. 8. The system ensures the validity of the data entered, creates a new account for the user and then redirects them to the Home Screen.
Alternative Flows:	1. The user enters some invalid data. 2. The system prompts the user to re - enter the correct information. 3. The correct information is then used to create the user account and then the user is redirected to the Login Screen.
Exceptions:	1. The entered email id is already a registered user in the system. 2. The system prompts the user that there is a pre - existing account registered and redirects them to the Login Screen.
Includes:	Redirect to Home Screen
Special Requirements:	N/A
Assumptions:	The user has an active internet connection
Notes and Issues:	Nil

6.3 Login Screen

Table 3: Login Screen: Use Case Description

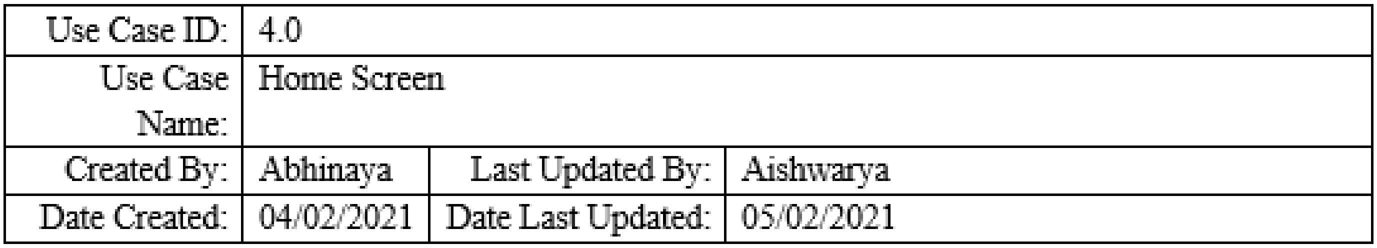


Use Case ID:	3.0		
Use Case Name:	Login		
Created By:	Abhinaya	Last Updated By:	Abhinaya
Date Created:	04/02/2021	Date Last Updated:	11/04/2021

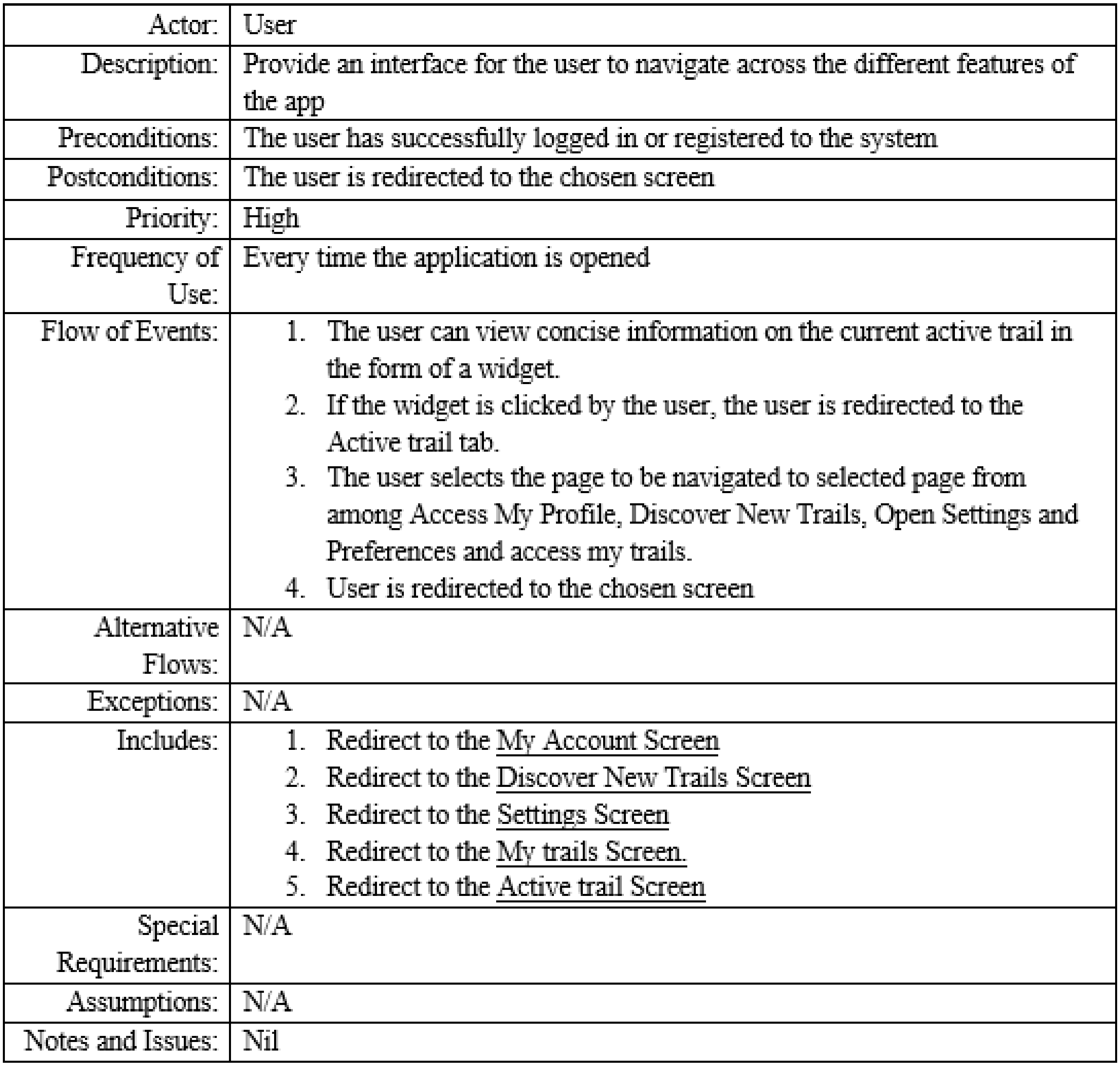
Actor:	User, Account System, Email Client
Description:	Executes the login procedure when the User Details are input
Preconditions:	1. Application must be started. 2. The user has registered with the application
Postconditions:	1. The user is logged on and moved the Home Screen. 2. Stored information of the user is retrieved
Priority:	High
Frequency of Use:	Every time the application is opened
Flow of Events:	1. The user enters email. 2. The user enters password. 3. Account System verifies the user information. 4. User is redirected to the Home Screen
Alternative Flows:	1. The user enters incorrect login information. 2. The state of the system does not change. 3. The user is prompted to re - enter details to Login Screen
Exceptions:	1. The user forgets login credentials. 2. The user can click forgot password. 3. An email is sent to the user to reset their password
Includes:	1. Redirect to Home Screen
Special Requirements:	N/A
Assumptions:	1. The user has an active internet connection
Notes and Issues:	Nil

6.4 Home Screen

Table 4: Home Screen: Use Case Description



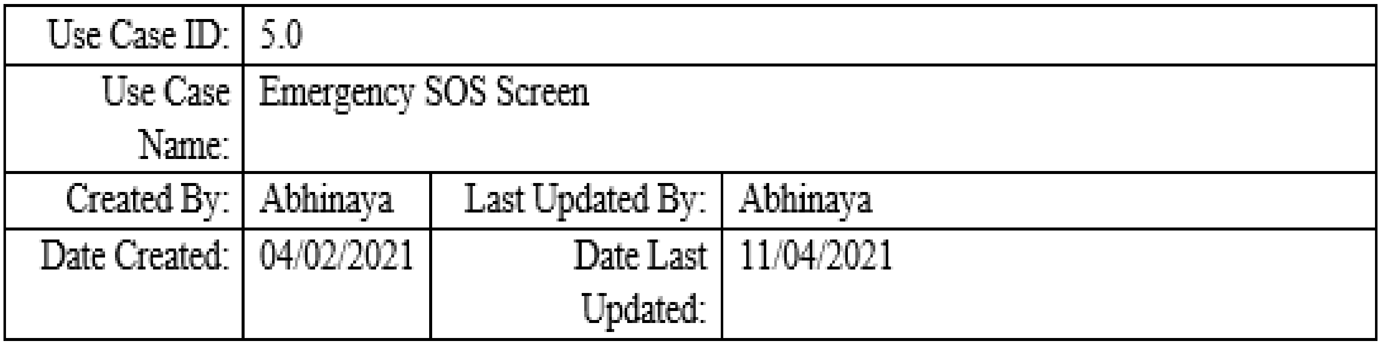
Use Case ID:	4.0		
Use Case Name:	Home Screen		
Created By:	Abhinaya	Last Updated By:	Aishwarya
Date Created:	04/02/2021	Date Last Updated:	05/02/2021



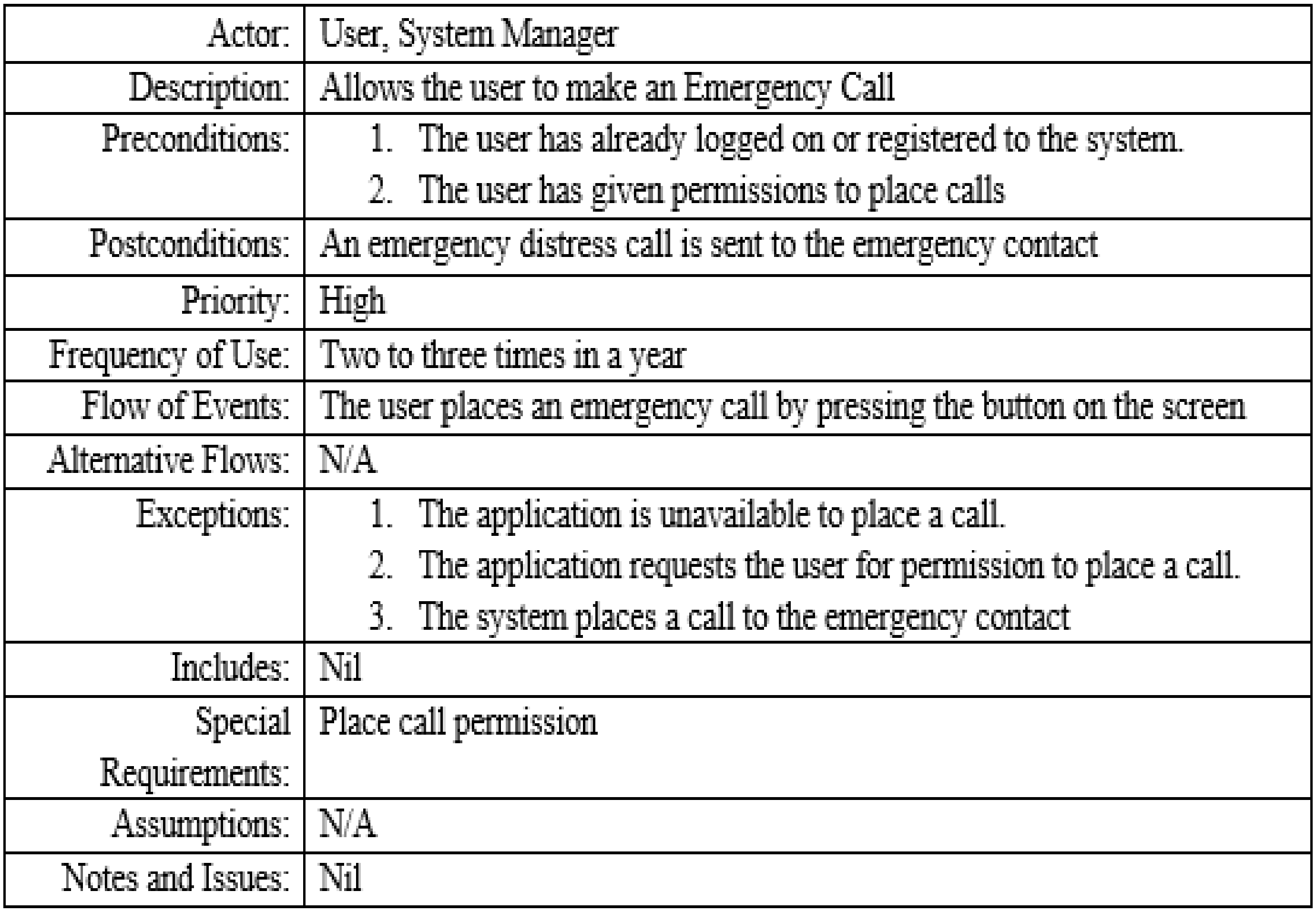
Actor:	User
Description:	Provide an interface for the user to navigate across the different features of the app
Preconditions:	The user has successfully logged in or registered to the system
Postconditions:	The user is redirected to the chosen screen
Priority:	High
Frequency of Use:	Every time the application is opened
Flow of Events:	1. The user can view concise information on the current active trail in the form of a widget. 2. If the widget is clicked by the user, the user is redirected to the Active trail tab. 3. The user selects the page to be navigated to selected page from among Access My Profile, Discover New Trails, Open Settings and Preferences and access my trails. 4. User is redirected to the chosen screen
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	1. Redirect to the My Account Screen 2. Redirect to the Discover New Trails Screen 3. Redirect to the Settings Screen 4. Redirect to the My trails Screen. 5. Redirect to the Active trail Screen
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	Nil

6.5 Emergency SOS Screen

Table 5: Emergency SOS Screen: Use Case Description



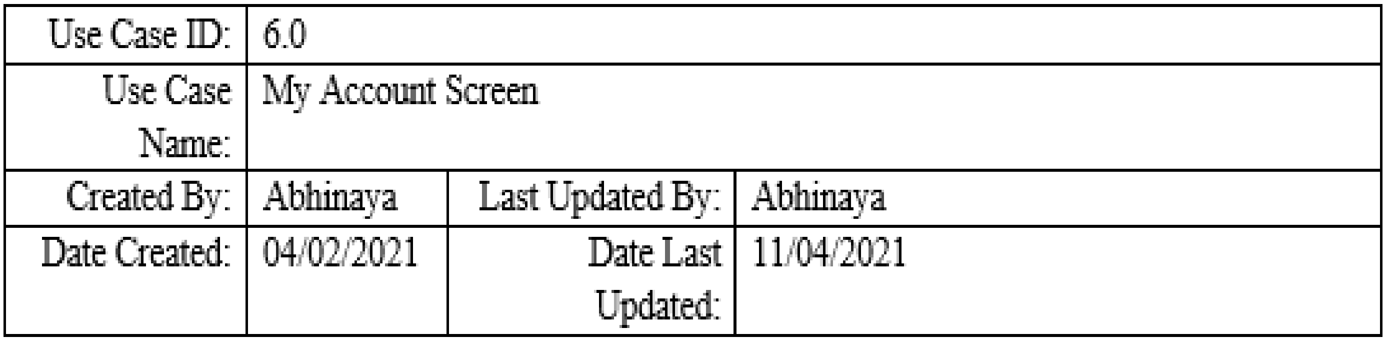
Use Case ID:	5.0		
Use Case Name:	Emergency SOS Screen		
Created By:	Abhinaya	Last Updated By:	Abhinaya
Date Created:	04/02/2021	Date Last Updated:	11/04/2021



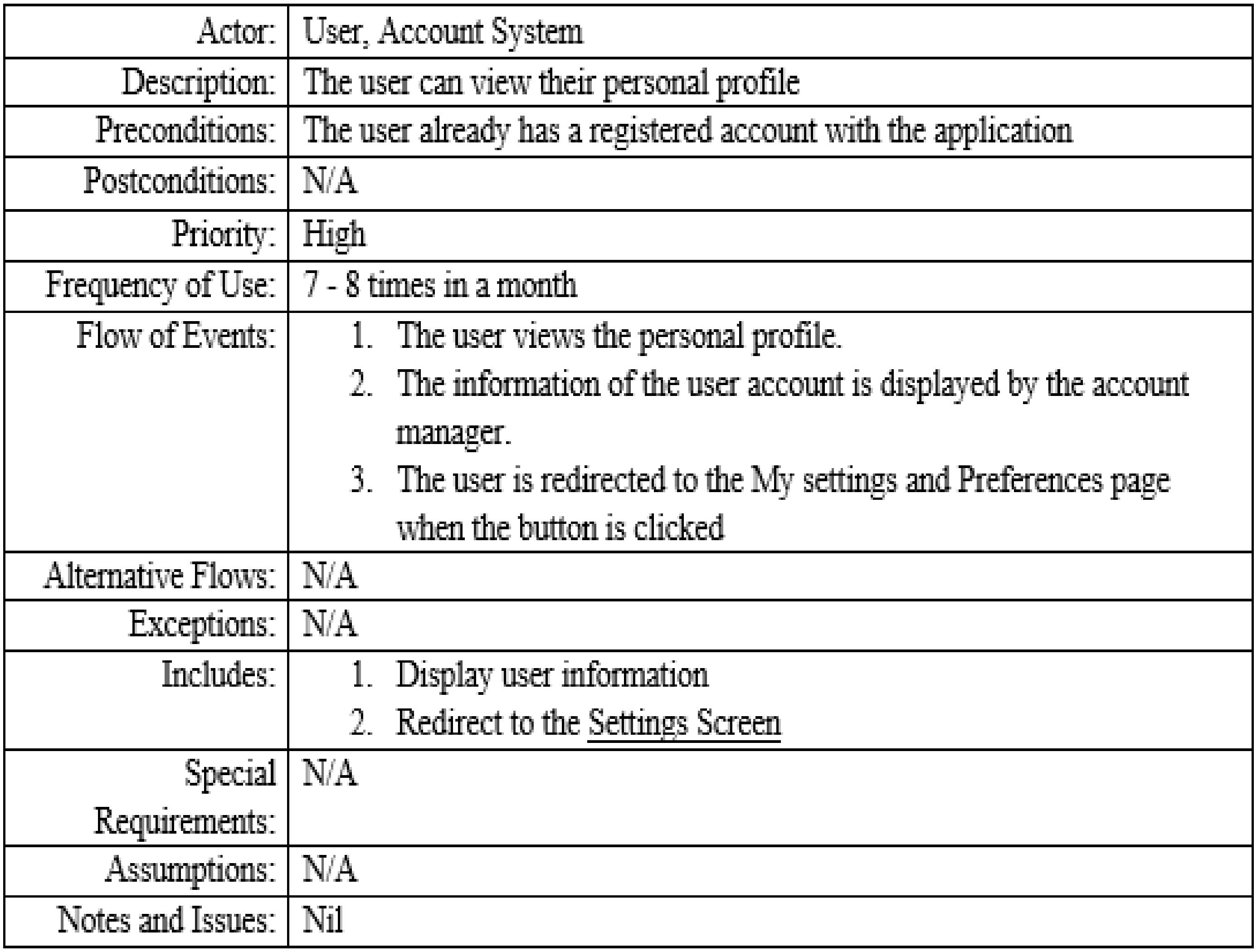
Actor:	User, System Manager
Description:	Allows the user to make an Emergency Call
Preconditions:	1. The user has already logged on or registered to the system. 2. The user has given permissions to place calls
Postconditions:	An emergency distress call is sent to the emergency contact
Priority:	High
Frequency of Use:	Two to three times in a year
Flow of Events:	The user places an emergency call by pressing the button on the screen
Alternative Flows:	N/A
Exceptions:	1. The application is unavailable to place a call. 2. The application requests the user for permission to place a call. 3. The system places a call to the emergency contact
Includes:	Nil
Special Requirements:	Place call permission
Assumptions:	N/A
Notes and Issues:	Nil

6.6 My Account Screen

Table 6: My Account Screen: Use Case Description



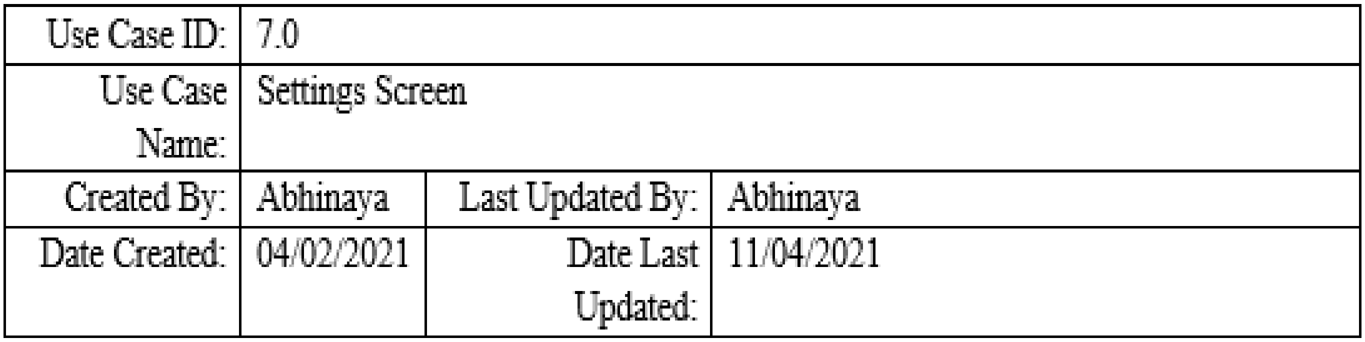
Use Case ID:	6.0		
Use Case Name:	My Account Screen		
Created By:	Abhinaya	Last Updated By:	Abhinaya
Date Created:	04/02/2021	Date Last Updated:	11/04/2021



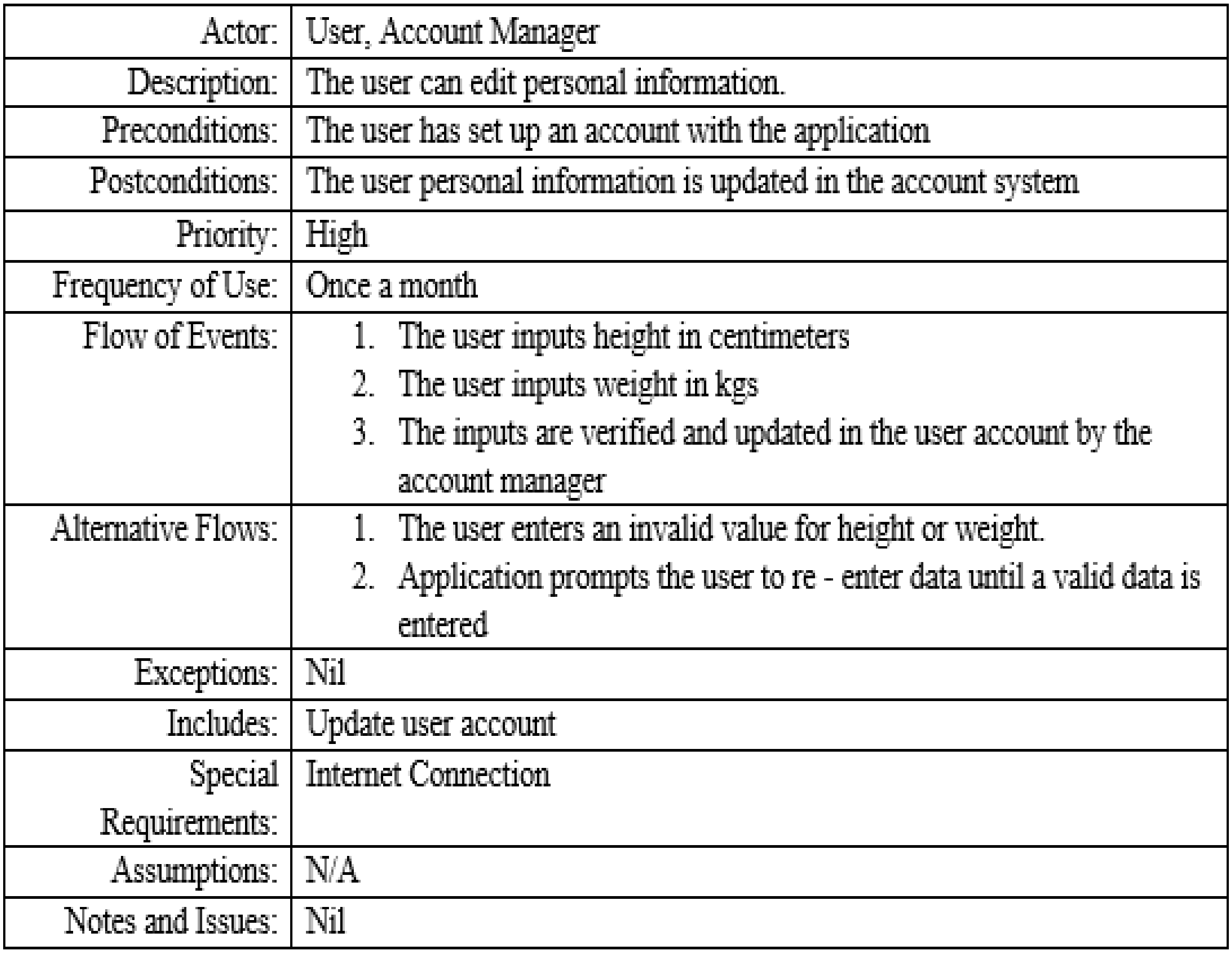
Actor:	User, Account System
Description:	The user can view their personal profile
Preconditions:	The user already has a registered account with the application
Postconditions:	N/A
Priority:	High
Frequency of Use:	7 - 8 times in a month
Flow of Events:	1. The user views the personal profile. 2. The information of the user account is displayed by the account manager. 3. The user is redirected to the My settings and Preferences page when the button is clicked
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	1. Display user information 2. Redirect to the Settings Screen
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	Nil

6.7 Settings Screen

Table 7: Settings Screen: Use Case Description



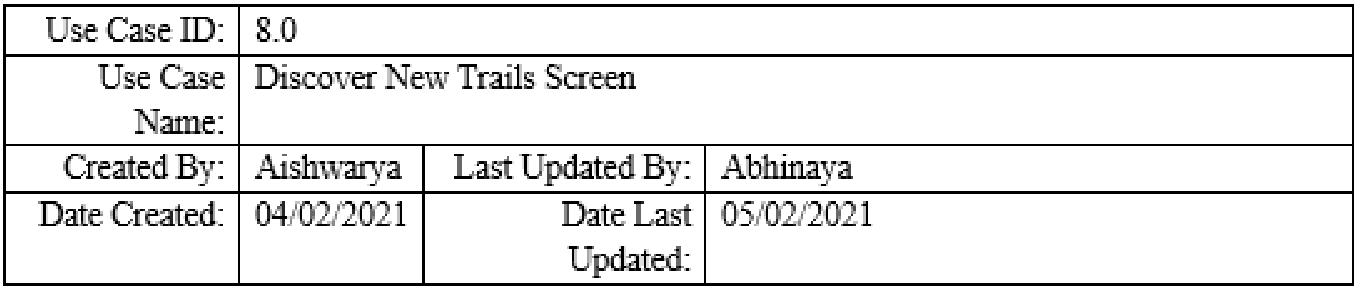
Use Case ID:	7.0		
Use Case Name:	Settings Screen		
Created By:	Abhinaya	Last Updated By:	Abhinaya
Date Created:	04/02/2021	Date Last Updated:	11/04/2021



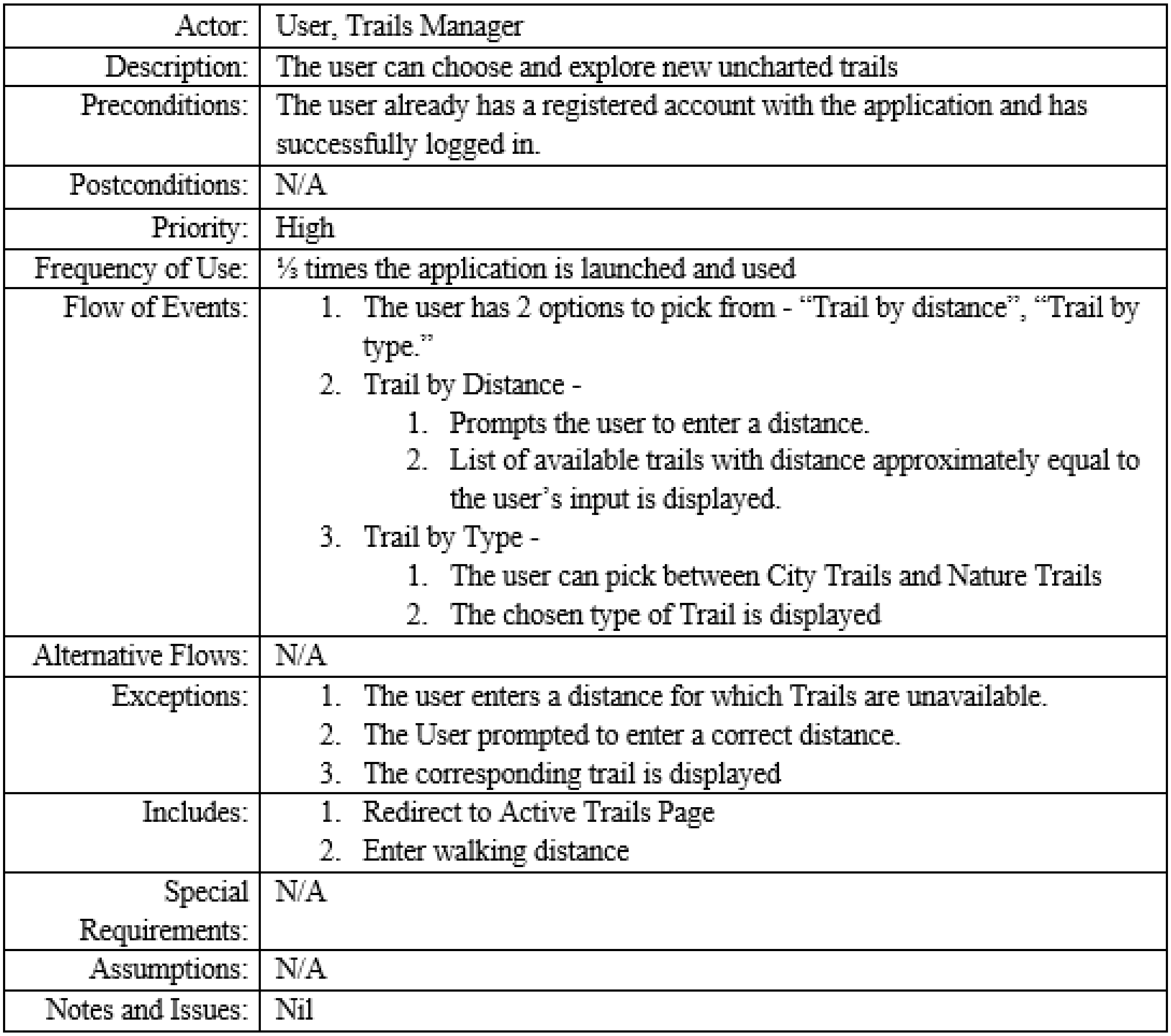
Actor:	User, Account Manager
Description:	The user can edit personal information.
Preconditions:	The user has set up an account with the application
Postconditions:	The user personal information is updated in the account system
Priority:	High
Frequency of Use:	Once a month
Flow of Events:	1. The user inputs height in centimeters 2. The user inputs weight in kgs 3. The inputs are verified and updated in the user account by the account manager
Alternative Flows:	1. The user enters an invalid value for height or weight. 2. Application prompts the user to re - enter data until a valid data is entered
Exceptions:	Nil
Includes:	Update user account
Special Requirements:	Internet Connection
Assumptions:	N/A
Notes and Issues:	Nil

6.8 Discover New Trails Screen

Table 8: Discover New Trails Screen: Use Case Description



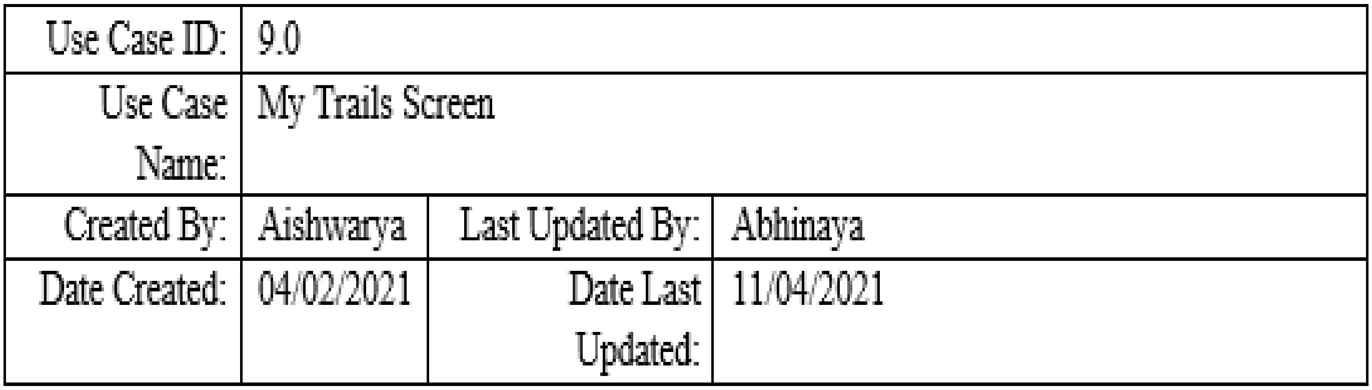
Use Case ID:	8.0		
Use Case Name:	Discover New Trails Screen		
Created By:	Aishwarya	Last Updated By:	Abhinaya
Date Created:	04/02/2021	Date Last Updated:	05/02/2021



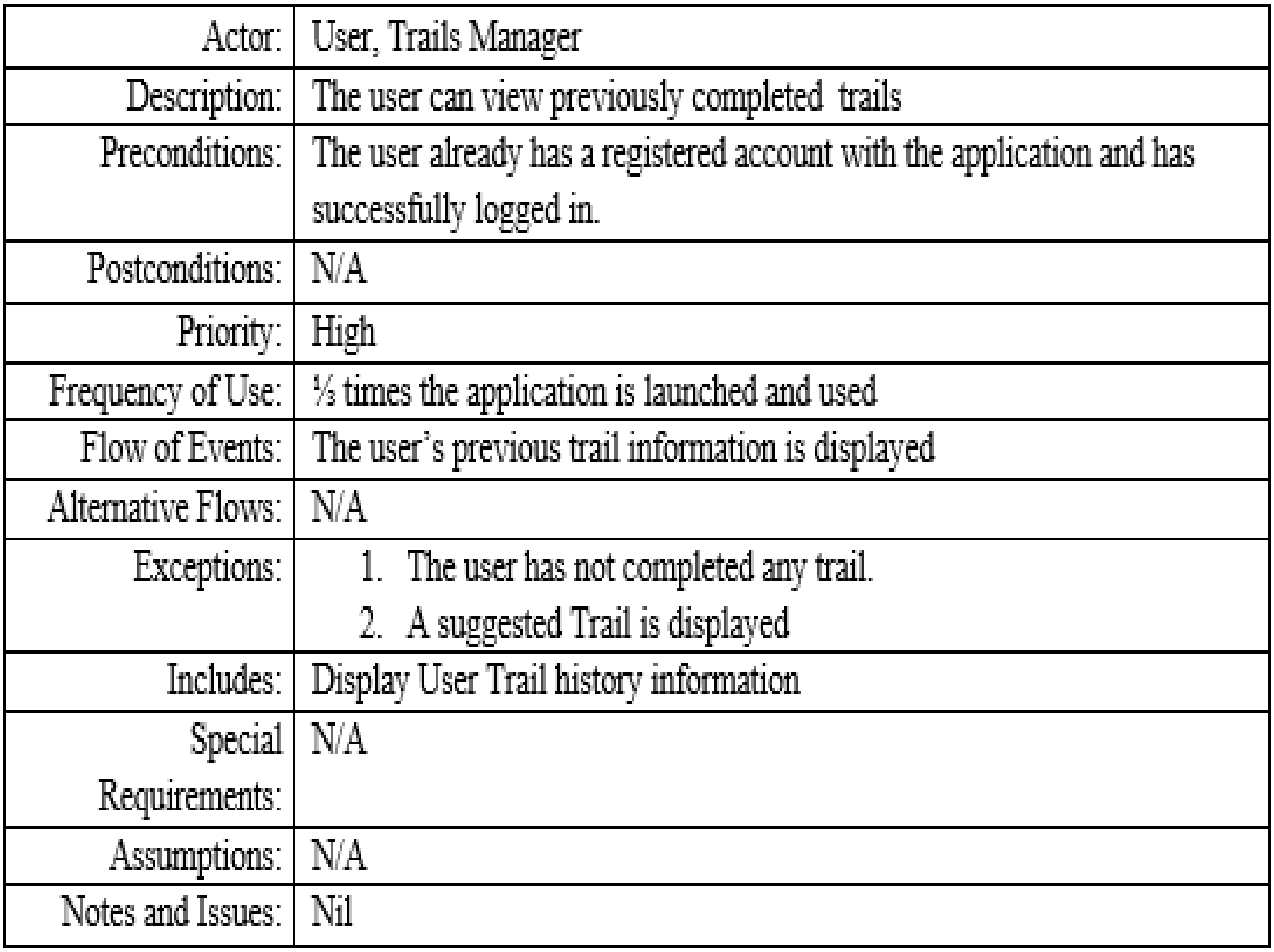
Actor:	User, Trails Manager
Description:	The user can choose and explore new uncharted trails
Preconditions:	The user already has a registered account with the application and has successfully logged in.
Postconditions:	N/A
Priority:	High
Frequency of Use:	⅓ times the application is launched and used
Flow of Events:	1. The user has 2 options to pick from - “Trail by distance”, “Trail by type.” 2. Trail by Distance - 1. Prompts the user to enter a distance. 2. List of available trails with distance approximately equal to the user’s input is displayed. 3. Trail by Type - 1. The user can pick between City Trails and Nature Trails 2. The chosen type of Trail is displayed
Alternative Flows:	N/A
Exceptions:	1. The user enters a distance for which Trails are unavailable. 2. The User prompted to enter a correct distance. 3. The corresponding trail is displayed
Includes:	1. Redirect to Active Trails Page 2. Enter walking distance
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	Nil

6.9 My Trails Screen

Table 9: My Trails Screen: Use Case Description



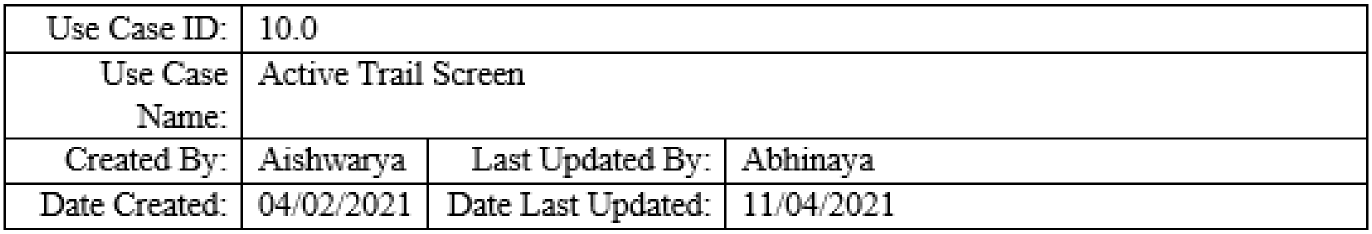
Use Case ID:	9.0		
Use Case Name:	My Trails Screen		
Created By:	Aishwarya	Last Updated By:	Abhinaya
Date Created:	04/02/2021	Date Last Updated:	11/04/2021



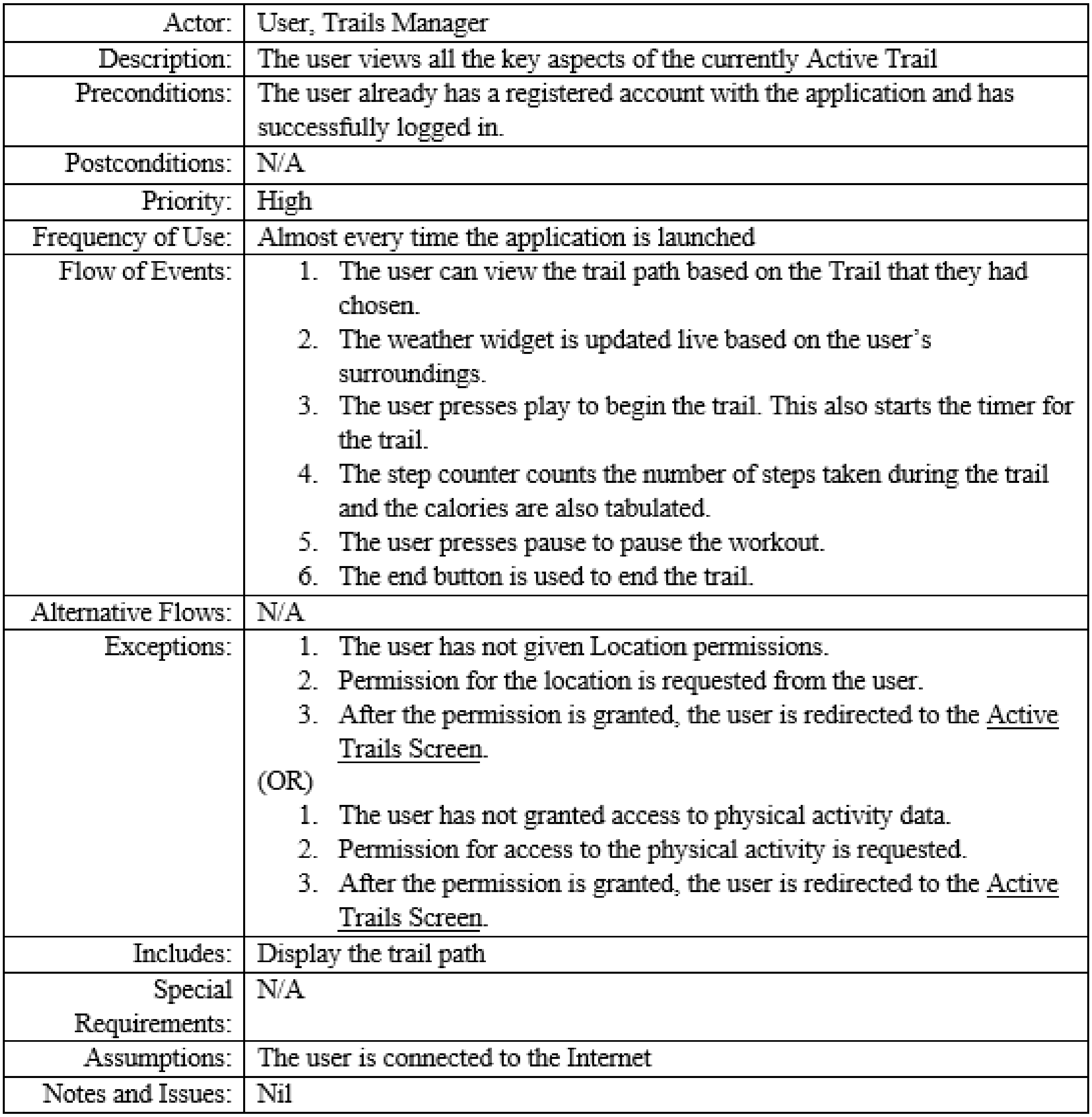
Actor:	User, Trails Manager
Description:	The user can view previously completed trails
Preconditions:	The user already has a registered account with the application and has successfully logged in.
Postconditions:	N/A
Priority:	High
Frequency of Use:	⅓ times the application is launched and used
Flow of Events:	The user's previous trail information is displayed
Alternative Flows:	N/A
Exceptions:	1. The user has not completed any trail. 2. A suggested Trail is displayed
Includes:	Display User Trail history information
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	Nil

6.10 Active Trail Screen

Table 10: Active Trail Screen: Use Case Description



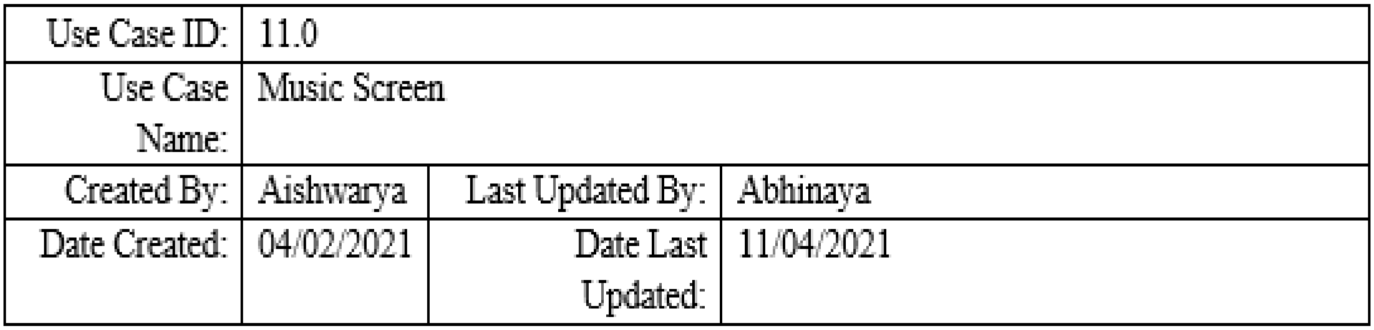
Use Case ID:	10.0		
Use Case Name:	Active Trail Screen		
Created By:	Aishwarya	Last Updated By:	Abhinaya
Date Created:	04/02/2021	Date Last Updated:	11/04/2021



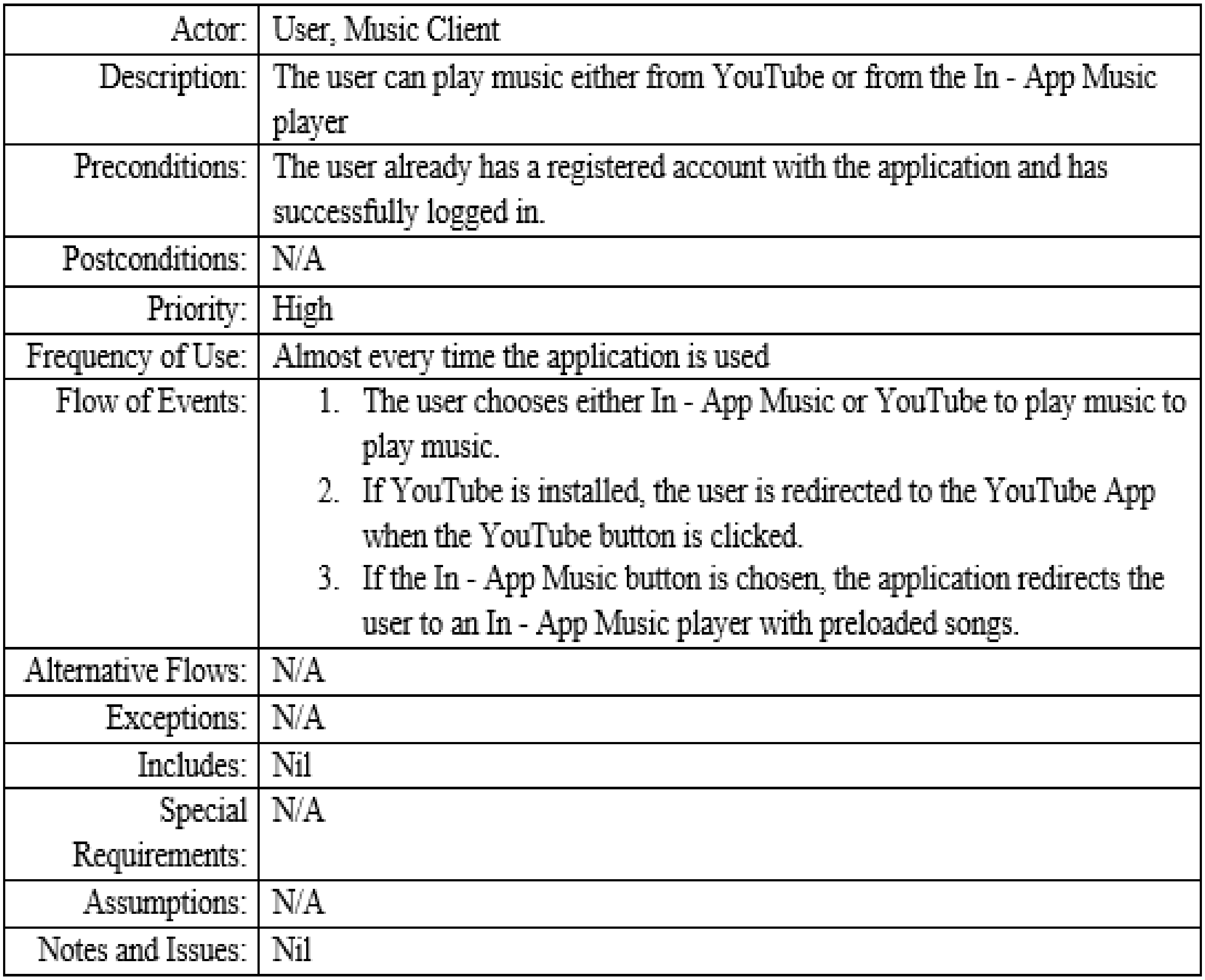
Actor:	User, Trails Manager
Description:	The user views all the key aspects of the currently Active Trail
Preconditions:	The user already has a registered account with the application and has successfully logged in.
Postconditions:	N/A
Priority:	High
Frequency of Use:	Almost every time the application is launched
Flow of Events:	1. The user can view the trail path based on the Trail that they had chosen. 2. The weather widget is updated live based on the user's surroundings. 3. The user presses play to begin the trail. This also starts the timer for the trail. 4. The step counter counts the number of steps taken during the trail and the calories are also tabulated. 5. The user presses pause to pause the workout. 6. The end button is used to end the trail.
Alternative Flows:	N/A
Exceptions:	1. The user has not given Location permissions. 2. Permission for the location is requested from the user. 3. After the permission is granted, the user is redirected to the Active Trails Screen. (OR) 1. The user has not granted access to physical activity data. 2. Permission for access to the physical activity is requested. 3. After the permission is granted, the user is redirected to the Active Trails Screen.
Includes:	Display the trail path
Special Requirements:	N/A
Assumptions:	The user is connected to the Internet
Notes and Issues:	Nil

6.11 Music Screen

Table 11: Music Screen: Use Case Description



Use Case ID:	11.0		
Use Case Name:	Music Screen		
Created By:	Aishwarya	Last Updated By:	Abhinaya
Date Created:	04/02/2021	Date Last Updated:	11/04/2021



Actor:	User, Music Client
Description:	The user can play music either from YouTube or from the In - App Music player
Preconditions:	The user already has a registered account with the application and has successfully logged in.
Postconditions:	N/A
Priority:	High
Frequency of Use:	Almost every time the application is used
Flow of Events:	1. The user chooses either In - App Music or YouTube to play music to play music. 2. If YouTube is installed, the user is redirected to the YouTube App when the YouTube button is clicked. 3. If the In - App Music button is chosen, the application redirects the user to an In - App Music player with preloaded songs.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	Nil
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	Nil

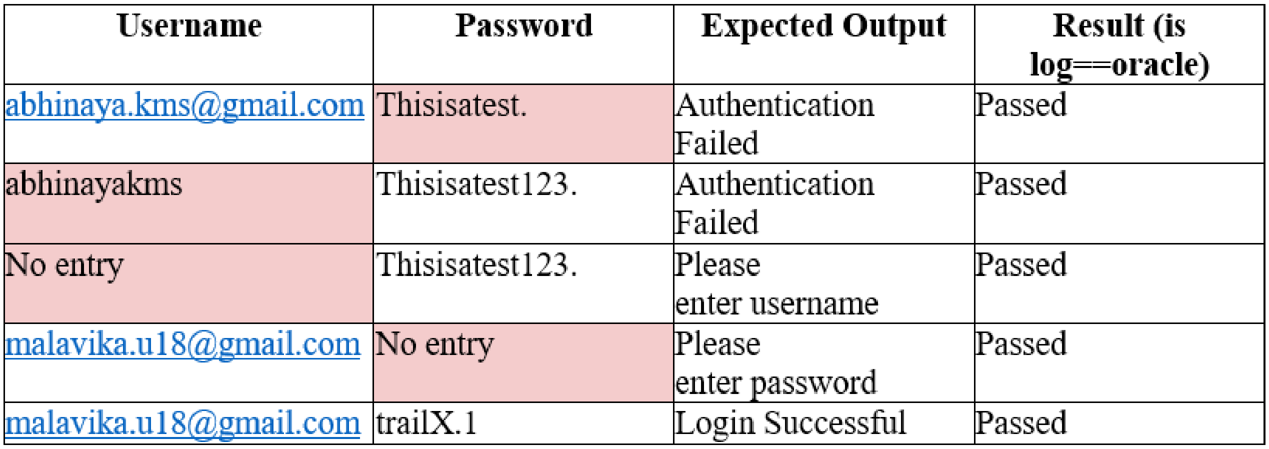
7. Testing

The two main types of testing used on our application are: Black and White Box testing. Both testing methods were conducted to analyze the system's capability of working to an acceptable value and to result in low-risk software. We did exhaustive manual testing of all features and aspects of the application system.

7.1 Black Box Testing

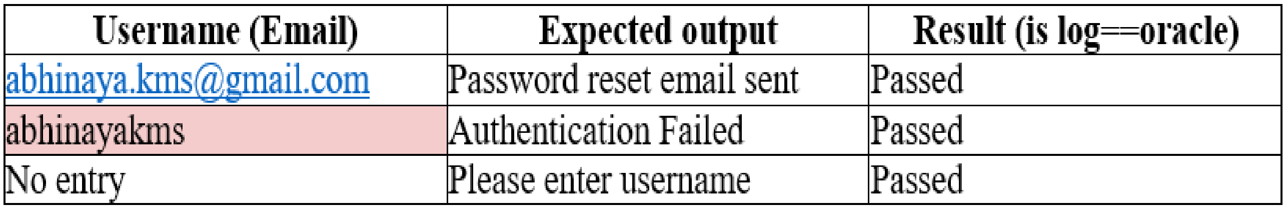
7.1.1 Login

Table 12: Test Case 1 – Checking if email and password are correct.



Username	Password	Expected Output	Result (is log==oracle)
abhinaya.kms@gmail.com	Thisisatest.	Authentication Failed	Passed
abhinayakms	Thisisatest123.	Authentication Failed	Passed
No entry	Thisisatest123.	Please enter username	Passed
malavika.u18@gmail.com	No entry	Please enter password	Passed
malavika.u18@gmail.com	trailX.1	Login Successful	Passed

Table 13: Test Case 2 – Forgot Password



Username (Email)	Expected output	Result (is log==oracle)
abhinaya.kms@gmail.com	Password reset email sent	Passed
abhinayakms	Authentication Failed	Passed
No entry	Please enter username	Passed

7.1.2 Register

Table 13: Test Case 1 – Setting a new username.

Test Input- New Username	Expected output	Result (is log==oracle)
abhi	Enter valid data	Passed
abhinayakms$	Enter valid data	Passed
abhinaya.kms	Enter valid data	Passed
abhinaya.kms@gmail.com	Username set!	Passed

Table 14: Test Case 2 – Setting a new password.

Test Input- New Password	Expected output	Result (is log==oracle)
123	Enter valid data	Passed
trailx	Enter valid data	Passed
trailX	Enter valid data	Passed
trailX1	Enter valid data	Passed
trailX1!	Enter valid data	Passed
trailX.1	Successful!	Passed

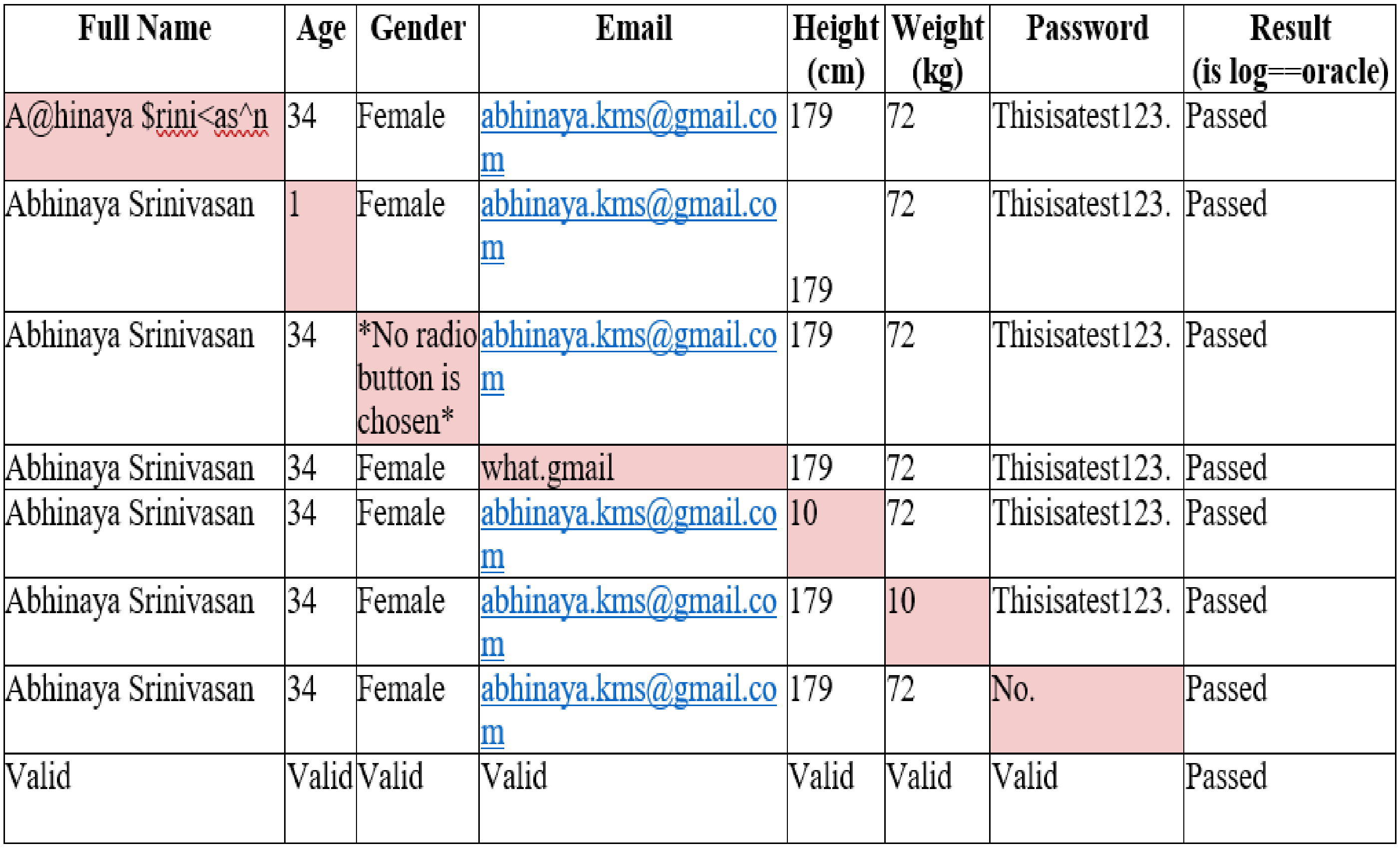
7.1.3 Discover New Trails - Distance by trails

Table 15: Test Case – Distance should be less than 10

Test Input- Distance in km (x)	Expected Output	Result (is log==oracle)
-1 (x<0)	No Trails Found	Passed
0.5 (0<x<1)	Trail Found!	Passed
3 (2<x<5)	Trail Found!	Passed
8 (6<x<10)	Trail Found!	Passed
45 (x>10)	No Trails Found	Passed

7.1.4 Settings

Table 16: Test Case – All edit details

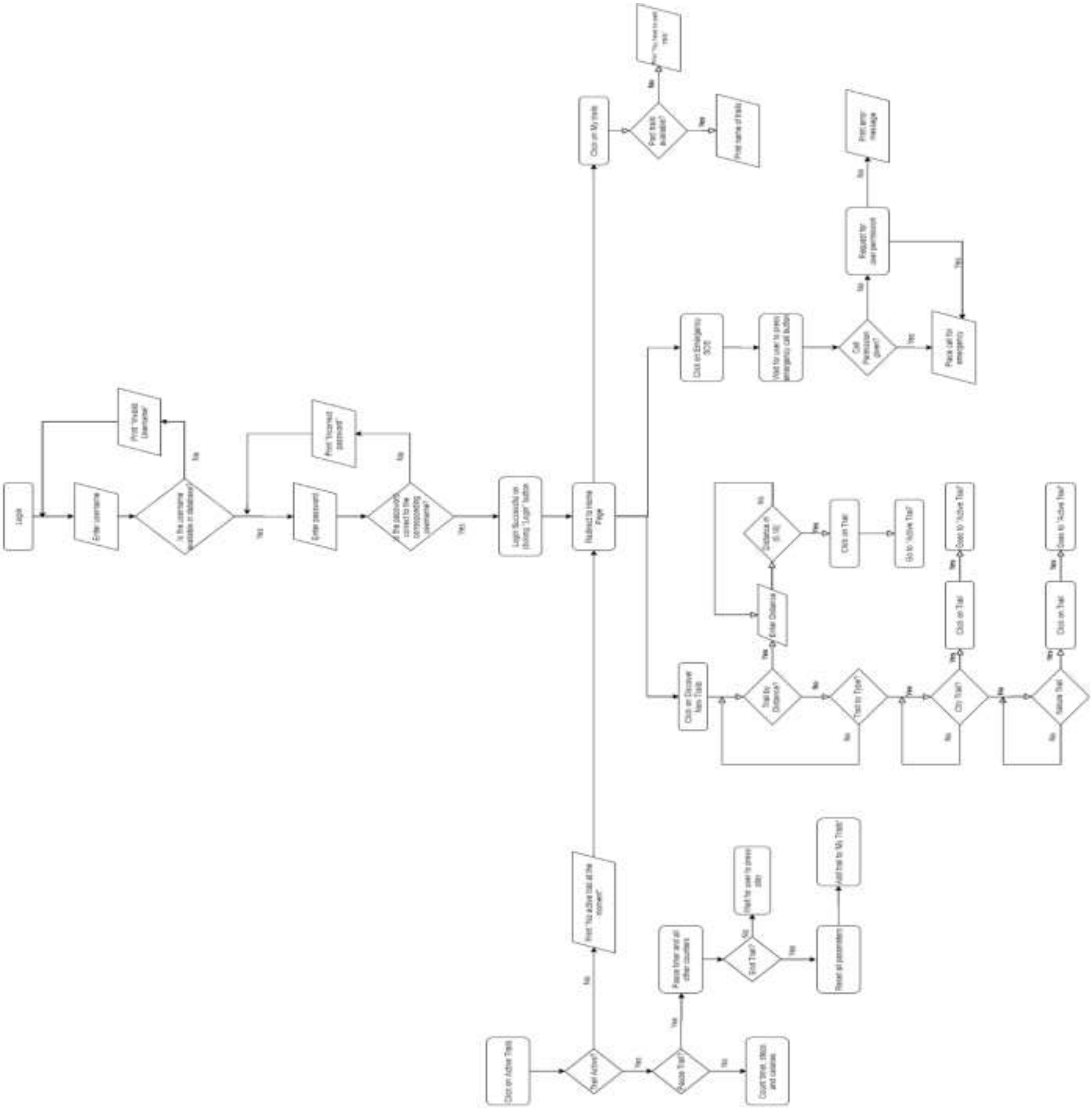


Full Name	Age	Gender	Email	Height (cm)	Weight (kg)	Password	Result (is log==oracle)
A@hinaya $rini<as^n	34	Female	abhinaya.kms@gmail.com	179	72	Thisisatest123.	Passed
Abhinaya Srinivasan	1	Female	abhinaya.kms@gmail.com	179	72	Thisisatest123.	Passed
Abhinaya Srinivasan	34	*No radio button is chosen*	abhinaya.kms@gmail.com	179	72	Thisisatest123.	Passed
Abhinaya Srinivasan	34	Female	what.gmail	179	72	Thisisatest123.	Passed
Abhinaya Srinivasan	34	Female	abhinaya.kms@gmail.com	10	72	Thisisatest123.	Passed
Abhinaya Srinivasan	34	Female	abhinaya.kms@gmail.com	179	10	Thisisatest123.	Passed
Abhinaya Srinivasan	34	Female	abhinaya.kms@gmail.com	179	72	No.	Passed
Valid	Valid	Valid	Valid	Valid	Valid	Valid	Passed

7.2 White Box Testing

The following is a control flow graph that depicts all possible paths of flow for the app. All paths have been tested during black box testing and other preliminary and post development testing processes.

Figure 12: White Box Testing



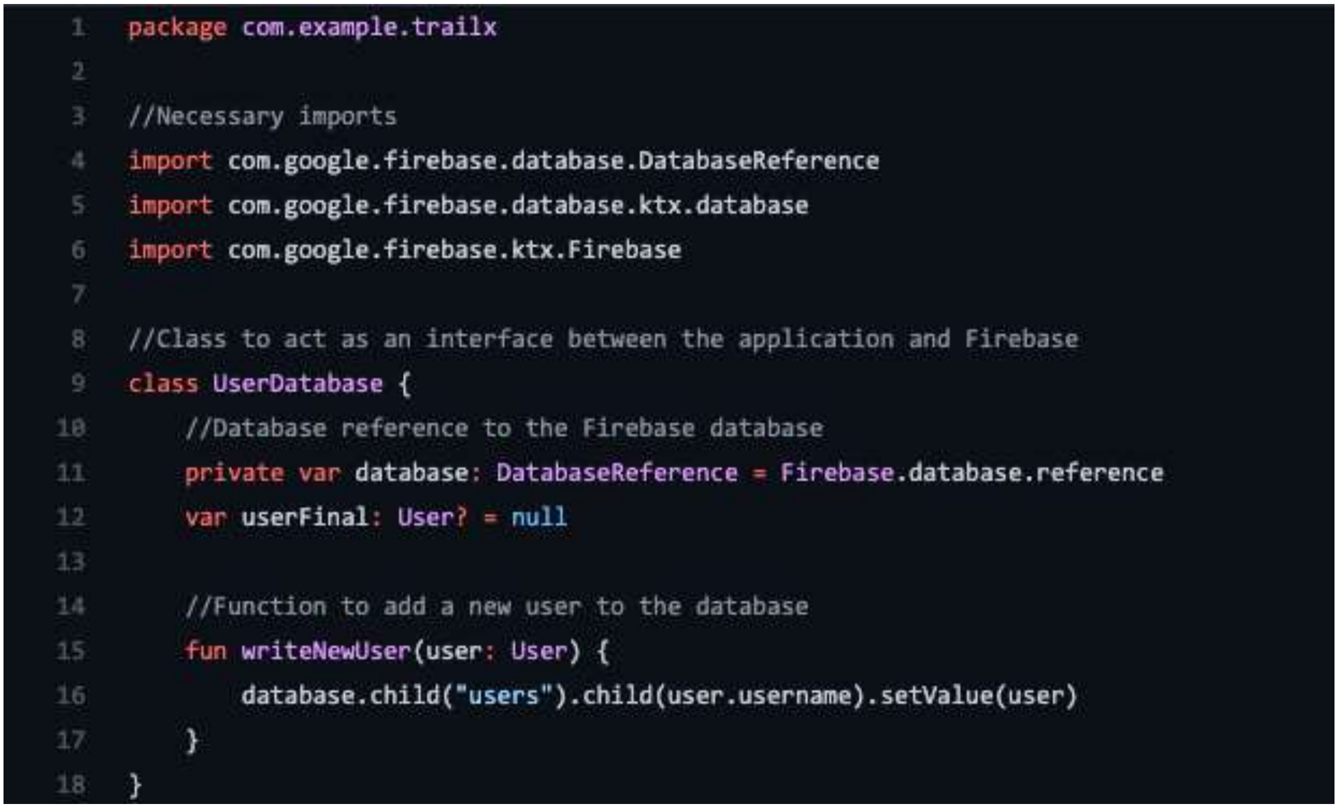
8. Design Pattern

8.1 Modularity

The application is implemented with the help of reusable classes and functions that are called in other contexts to prevent repetition and allow variety in use. This not only helps reduce load on certain classes that are already implemented, but also enables easy extendibility of the application in future.

For instance, the UserDatabase class contains the function writeNewUser(user: User), that is called in other classes such as SettingsScreen.

Figure 13: UserDatabase class Code Snippet

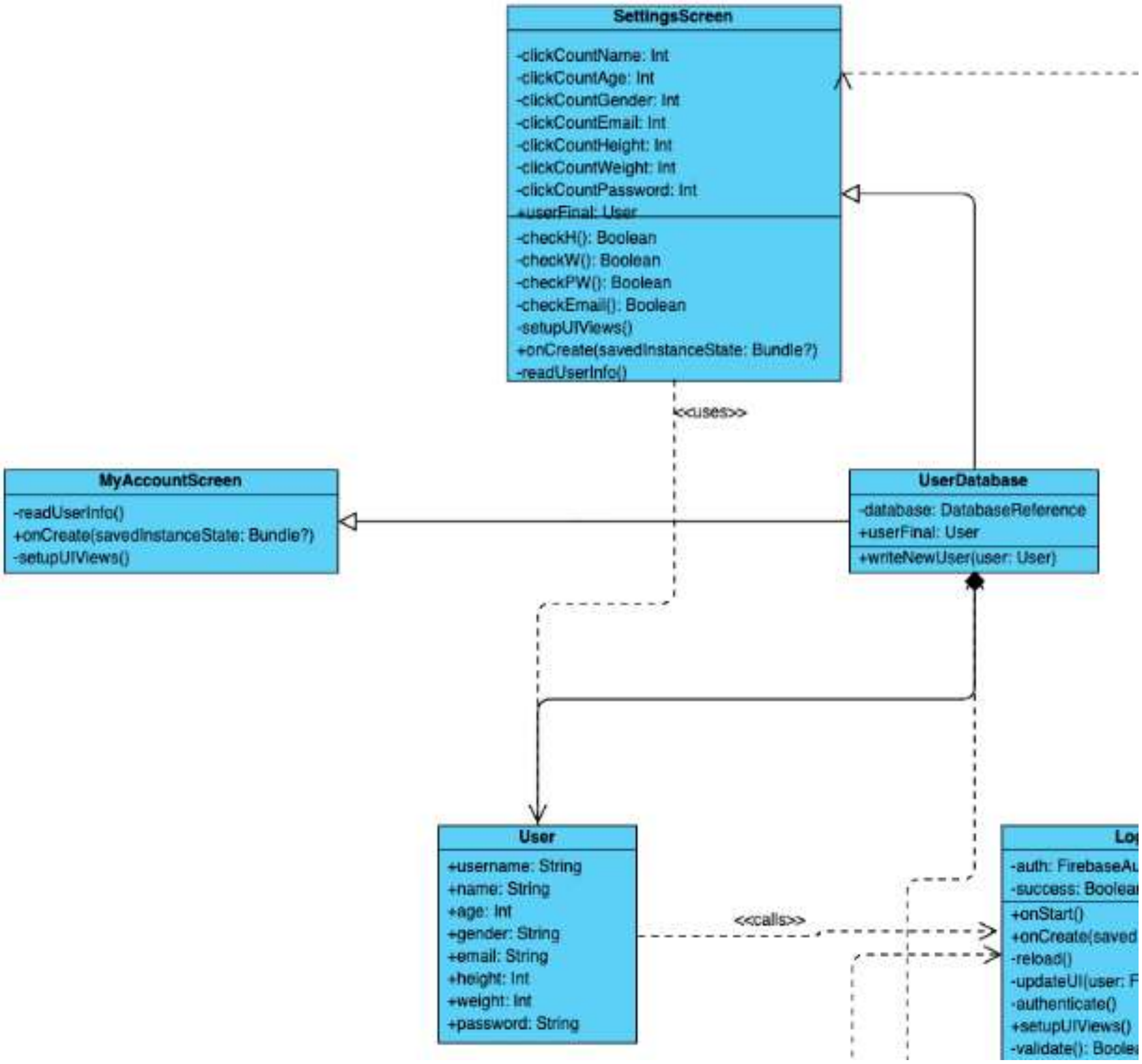


```
package com.example.trailx

//Necessary imports
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.ktx.database
import com.google.firebase.ktx.Firebase

//Class to act as an interface between the application and Firebase
class UserDatabase {
    //Database reference to the Firebase database
    private var database: DatabaseReference = Firebase.database.reference
    var userFinal: User? = null

    //Function to add a new user to the database
    fun writeNewUser(user: User) {
        database.child("users").child(user.username).setValue(user)
    }
}
```

8.2 Model View Control Pattern

To allow more flexibility in our application, we have made implemented classes to make code easier to access and print user data. This design makes every class independent, thus, allowing easy maintenance and future enhancements without affecting current data.

Figure 14: MVC Implementation



In our application, the following classes are in parallel with the MVC pattern:

Model: User.kt

View: MyAccountScreen.kt

Controller: SettingsScreen.kt

The MVC design pattern is demonstrated by the UserDatabase.kt class.

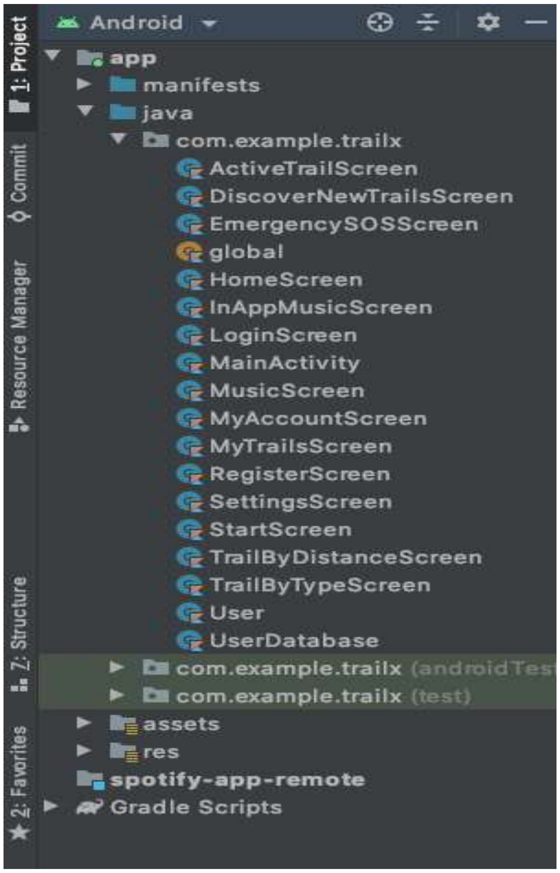
9. Software Design Principles

SOLID is a mnemonic acronym for the five object-oriented computer programming design principles intended to make a software application more flexible, reusable understandable and maintainable. Formulated by Robert C. Martin in a 2000 essay, "Design Principles and Design Patterns", the brand goal of these principles is to reduce dependencies so that engineers change one part of the software design easily without impacting another. In addition, these principles make a design easy to understand for fellow engineers and developers, thus helping to avoid issues and to build adaptive, effective, and agile software. Through the front-end and back-end of our application's software development, we have best implemented the following principles:

9.1 Single Responsibility Principle

While working on the application, we have carefully thought about future changes and additions to the application as requirements change over time. When a requirement changes, it causes a change in at least one of the classes. Therefore, the more the responsibilities a class has, the harder it is to implement this change. Keeping this in mind, our application's software is designed such that every class has one and only one responsibility. We have made at least one class for each of our use cases, thus reducing the number of responsibilities a single class has.

Figure 15: List of classes in the application

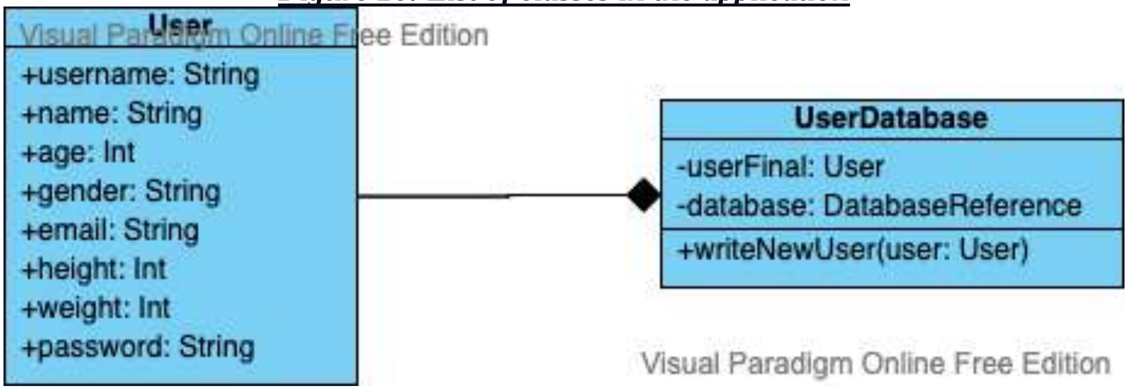


```
▼ app
    ▶ manifests
    ▼ java
        ▼ com.example.trailx
              ActiveTrailScreen
              DiscoverNewTrailsScreen
              EmergencySOSScreen
              global
              HomeScreen
              InAppMusicScreen
              LoginScreen
              MainActivity
              MusicScreen
              MyAccountScreen
              MyTrailsScreen
              RegisterScreen
              SettingsScreen
              StartScreen
              TrailByDistanceScreen
              TrailByTypeScreen
              User
              UserDatabase
        ▶ com.example.trailx (androidTes
        ▶ com.example.trailx (test)
    ▶ assets
    ▶ res
  spotify-app-remote
▶ Gradle Scripts
```

9.2 Open/Closed Principle

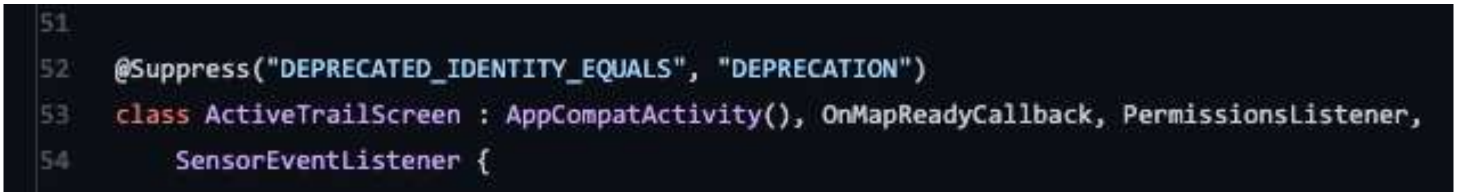
The idea of this principle is that we must be able to add new functions to our software design without modifying the existing code. In our *TrailX* application, we have implemented this principle through class composition.

Figure 16: List of classes in the application



This principle is also implemented through inheritance with the help of parent classes such as AppCompatActivity(), OnMapReadyCallback, PermissionsListener and SensorEventListener that are inherited by the child class ActiveTrailsScreen. Below is a code snippet indicating this inheritance.

Figure 17: Code Snippet of ActiveTrailsScreen

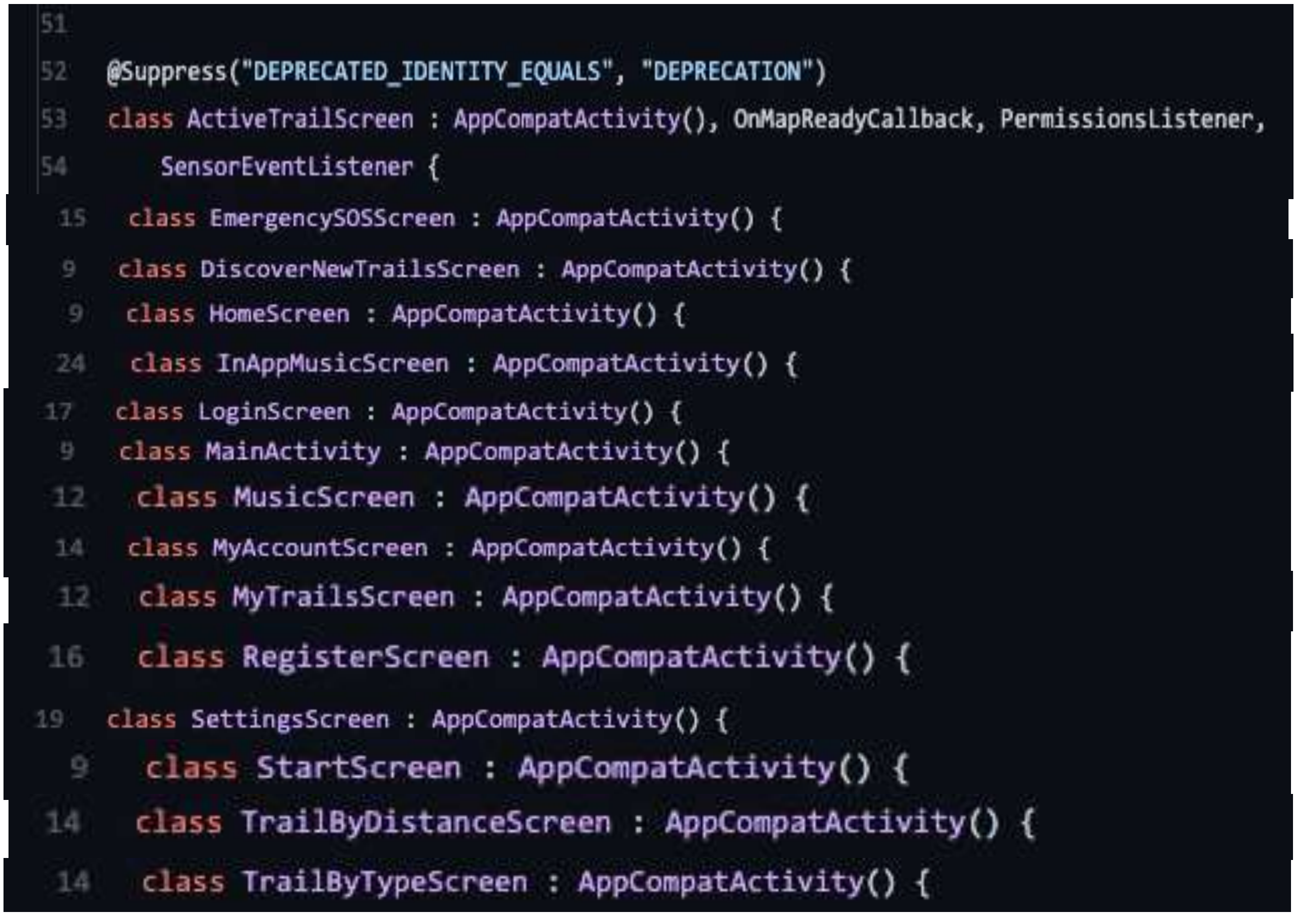


```

@Suppress("DEPRECATED_IDENTITY_EQUALS", "DEPRECATION")
class ActiveTrailScreen : AppCompatActivity(), OnMapReadyCallback, PermissionsListener,
    SensorEventListener {
```

9.3 Liskov Substitution Principle

The principle essentially states that the parent classes should be easily substituted with the child classes without overloading the application. In our software design, all the activities inherit AppCompatActivity and can therefore use all its functionalities. In this manner, the child classes can substitute without negatively affecting the application in any manner.

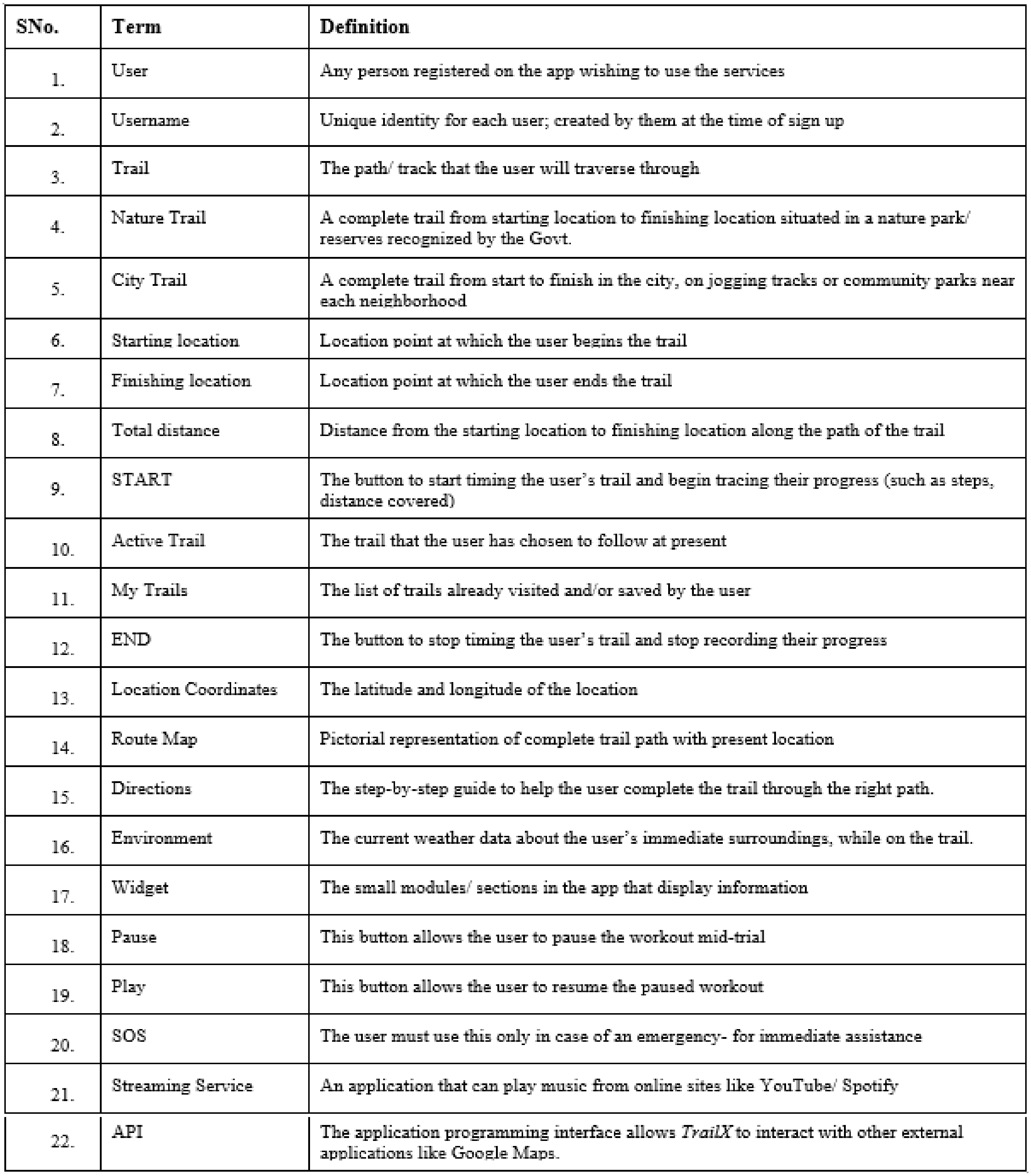
<u>*Figure 18: Code Snippets of classes from the application indicating inheritance.*</u>



```

@Suppress("DEPRECATED_IDENTITY_EQUALS", "DEPRECATION")
class ActiveTrailScreen : AppCompatActivity(), OnMapReadyCallback, PermissionsListener,
    SensorEventListener {
class EmergencySOSScreen : AppCompatActivity() {
class DiscoverNewTrailsScreen : AppCompatActivity() {
class HomeScreen : AppCompatActivity() {
class InAppMusicScreen : AppCompatActivity() {
class LoginScreen : AppCompatActivity() {
class MainActivity : AppCompatActivity() {
class MusicScreen : AppCompatActivity() {
class MyAccountScreen : AppCompatActivity() {
class MyTrailsScreen : AppCompatActivity() {
class RegisterScreen : AppCompatActivity() {
class SettingsScreen : AppCompatActivity() {
class StartScreen : AppCompatActivity() {
class TrailByDistanceScreen : AppCompatActivity() {
class TrailByTypeScreen : AppCompatActivity() {
```

Appendix A: Glossary Model

Table 18: Data Dictionary



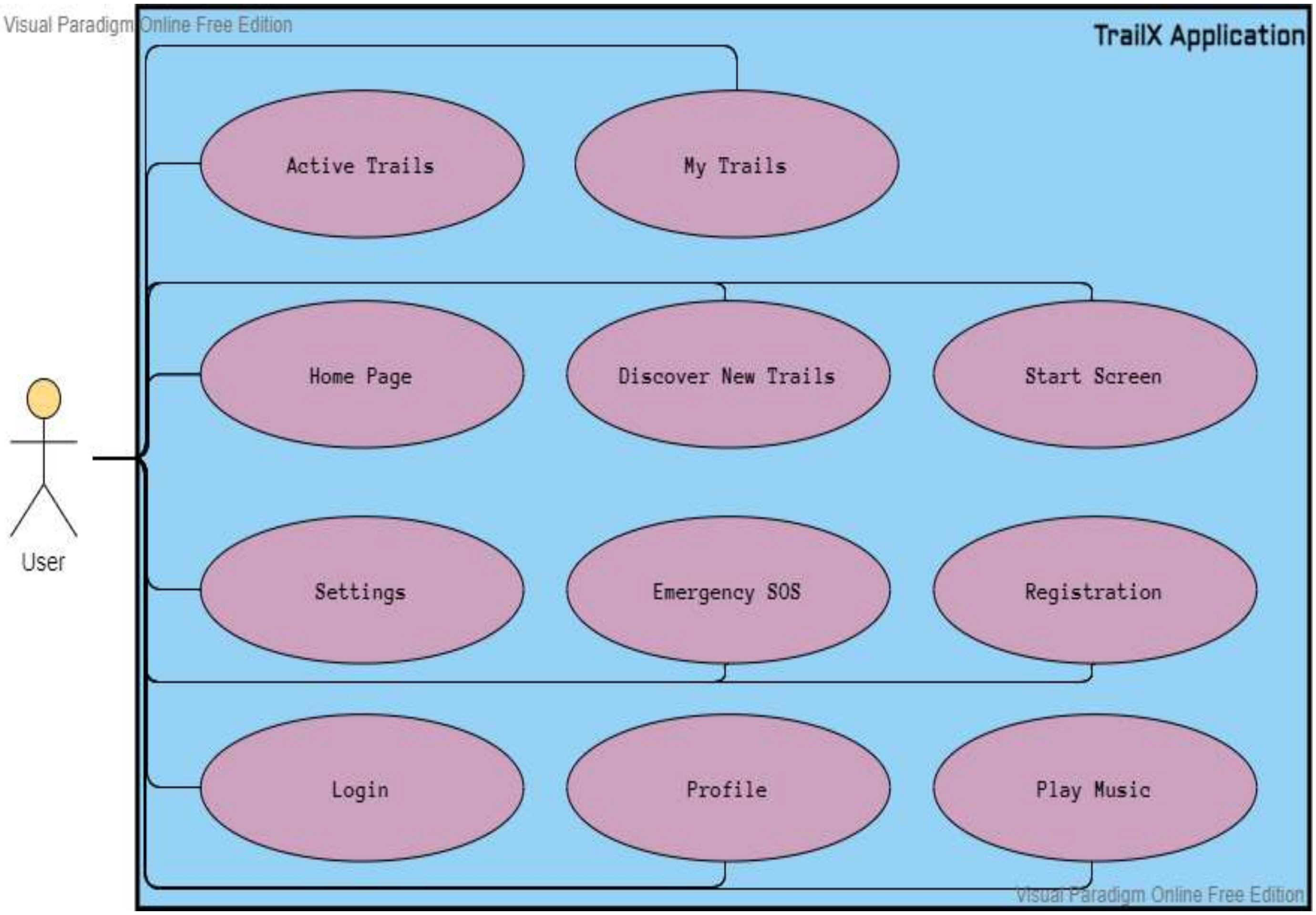
SNo.	Term	Definition
1.	User	Any person registered on the app wishing to use the services
2.	Username	Unique identity for each user; created by them at the time of sign up
3.	Trail	The path/ track that the user will traverse through
4.	Nature Trail	A complete trail from starting location to finishing location situated in a nature park/ reserves recognized by the Govt.
5.	City Trail	A complete trail from start to finish in the city, on jogging tracks or community parks near each neighborhood
6.	Starting location	Location point at which the user begins the trail
7.	Finishing location	Location point at which the user ends the trail
8.	Total distance	Distance from the starting location to finishing location along the path of the trail
9.	START	The button to start timing the user's trail and begin tracing their progress (such as steps, distance covered)
10.	Active Trail	The trail that the user has chosen to follow at present
11.	My Trails	The list of trails already visited and/or saved by the user
12.	END	The button to stop timing the user's trail and stop recording their progress
13.	Location Coordinates	The latitude and longitude of the location
14.	Route Map	Pictorial representation of complete trail path with present location
15.	Directions	The step-by-step guide to help the user complete the trail through the right path.
16.	Environment	The current weather data about the user's immediate surroundings, while on the trail.
17.	Widget	The small modules/ sections in the app that display information
18.	Pause	This button allows the user to pause the workout mid-trial
19.	Play	This button allows the user to resume the paused workout
20.	SOS	The user must use this only in case of an emergency- for immediate assistance
21.	Streaming Service	An application that can play music from online sites like YouTube/ Spotify
22.	API	The application programming interface allows *TrailX* to interact with other external applications like Google Maps.

Appendix B: Analysis Models

Use Case Diagrams

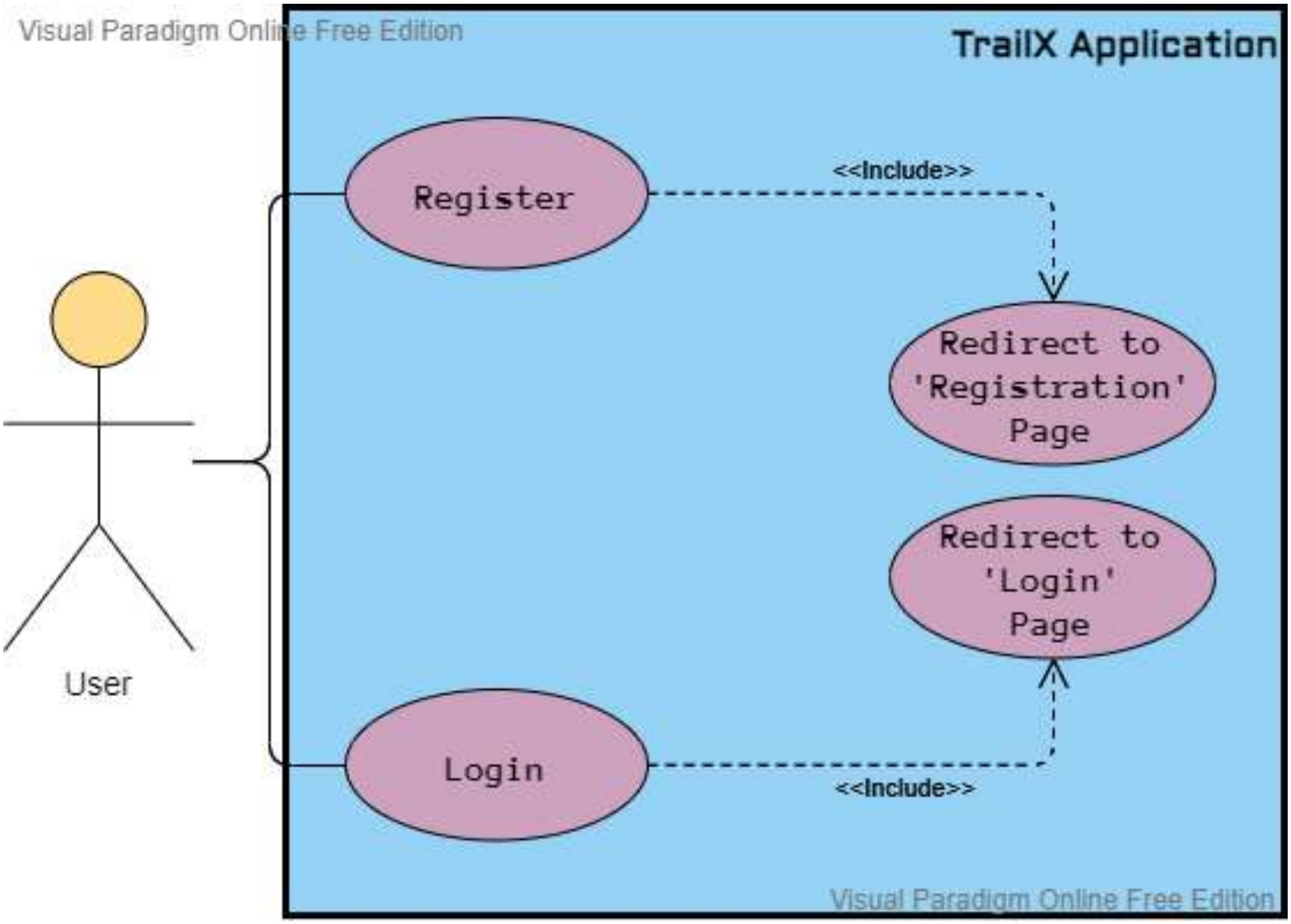
1. Complete Application Diagram

Figure 19: Complete Application – Use Case Diagram



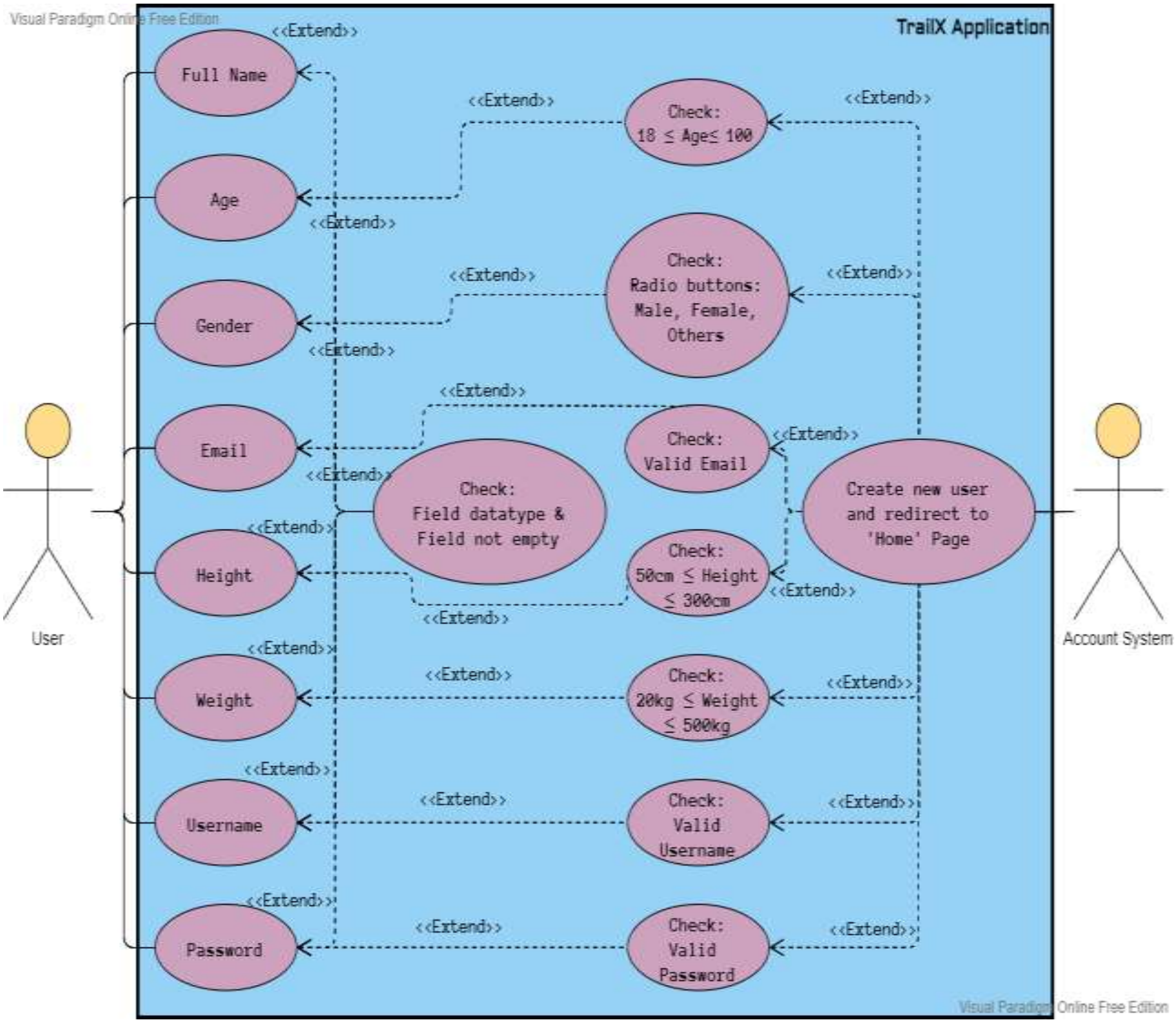
2. Start Screen

Figure 20: Start Screen – Use Case Diagram


Visual Paradigm Online Free Edition
TrailX Application
Register
<<Include>>
Redirect to 'Registration' Page
Redirect to 'Login' Page
User
Login
<<Include>>
Visual Paradigm Online Free Edition

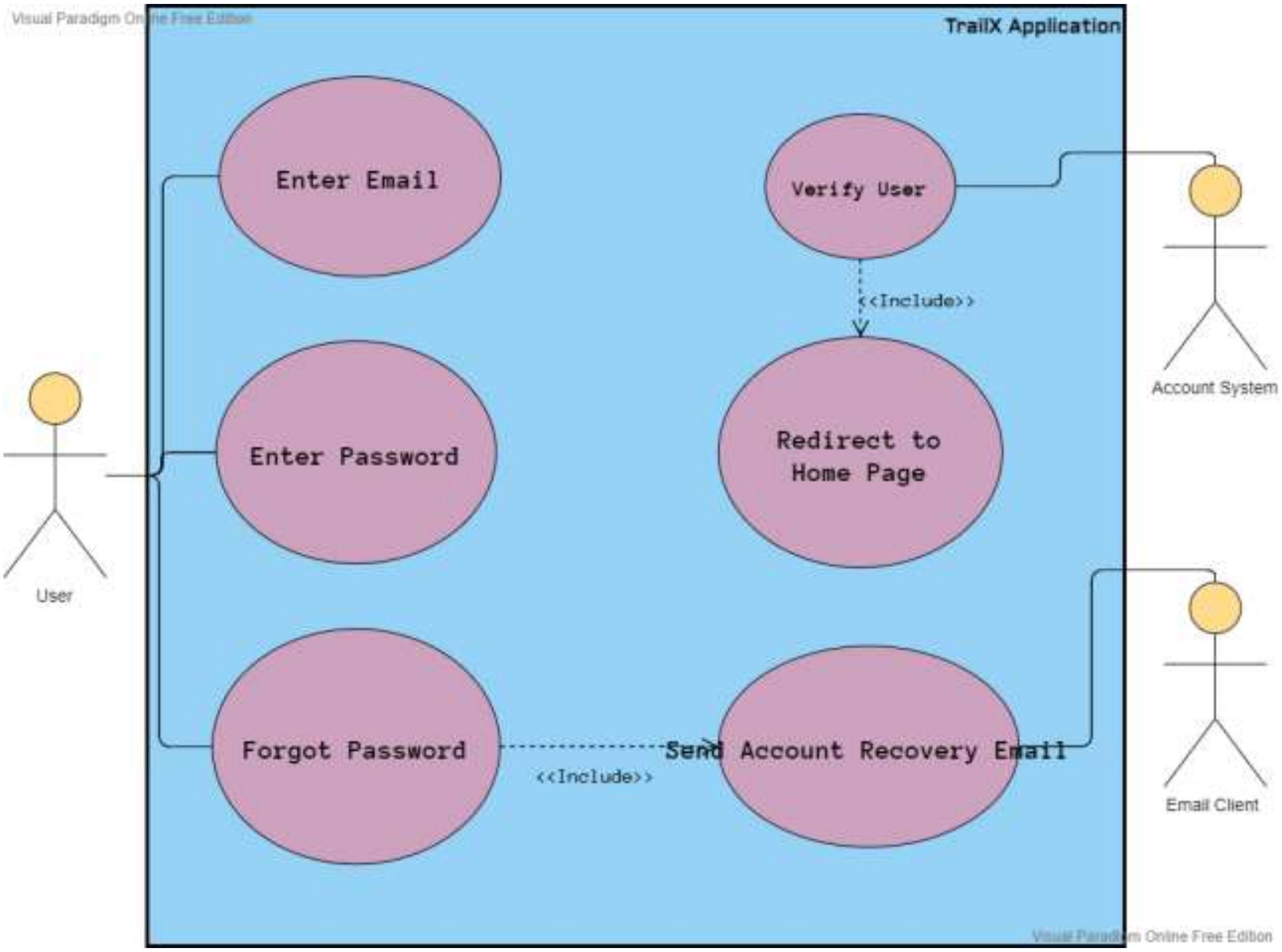
3. Registration Screen

Figure 21: Registration Screen – Use Case Diagram


Visual Paradigm Online Free Edition
TrailX Application
Full Name
Age
Gender
Email
Height
Weight
Username
Password
Check: 18 ≤ Age≤ 100
Check: Radio buttons: Male, Female, Others
Check: Valid Email
Check: Field datatype & Field not empty
Check: 50cm ≤ Height ≤ 300cm
Check: 20kg ≤ Weight ≤ 500kg
Check: Valid Username
Check: Valid Password
Create new user and redirect to 'Home' Page
<<Extend>>
User
Account System
Visual Paradigm Online Free Edition

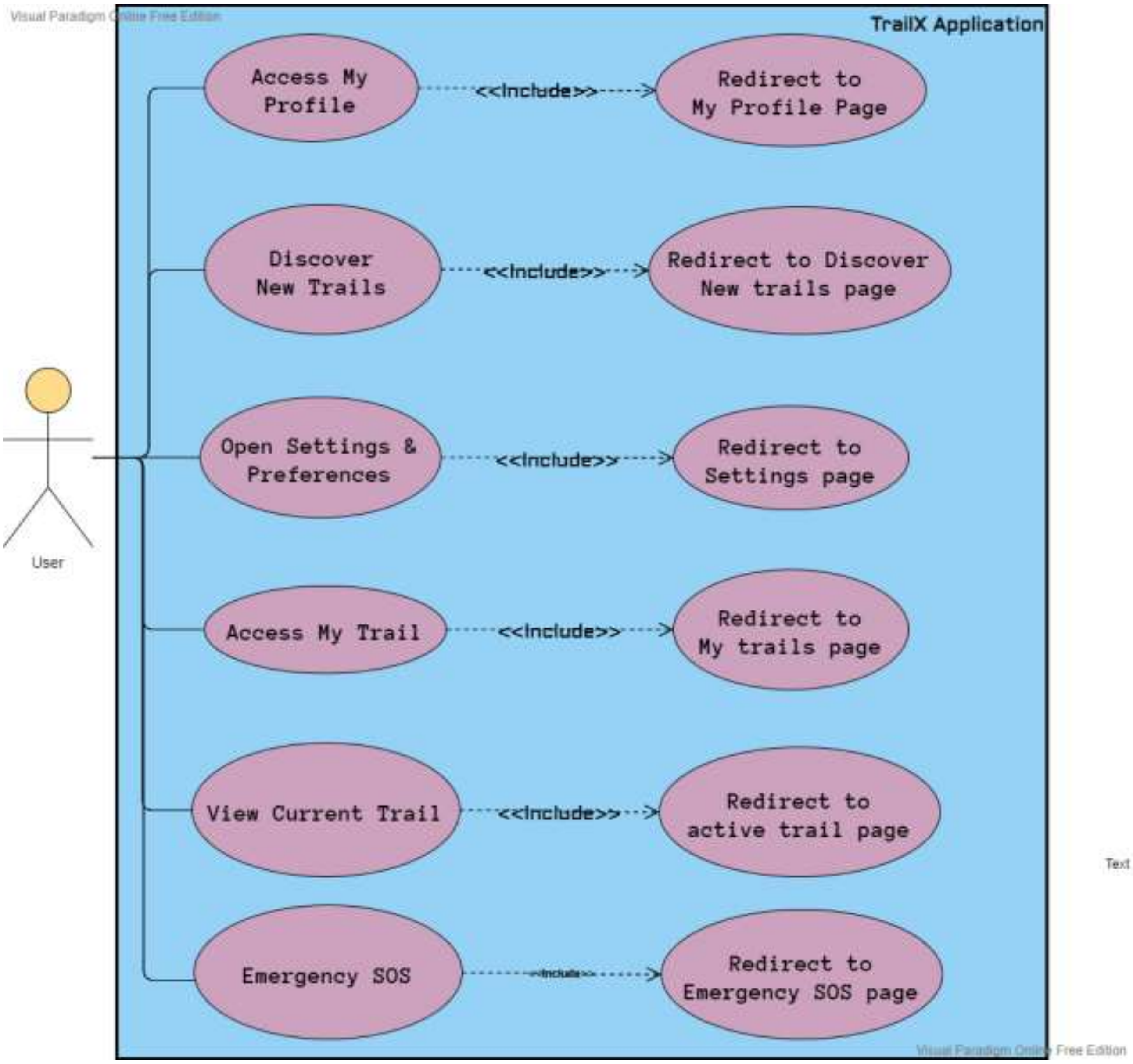
4. Login Screen

Figure 22: Login Screen – Use Case Diagram



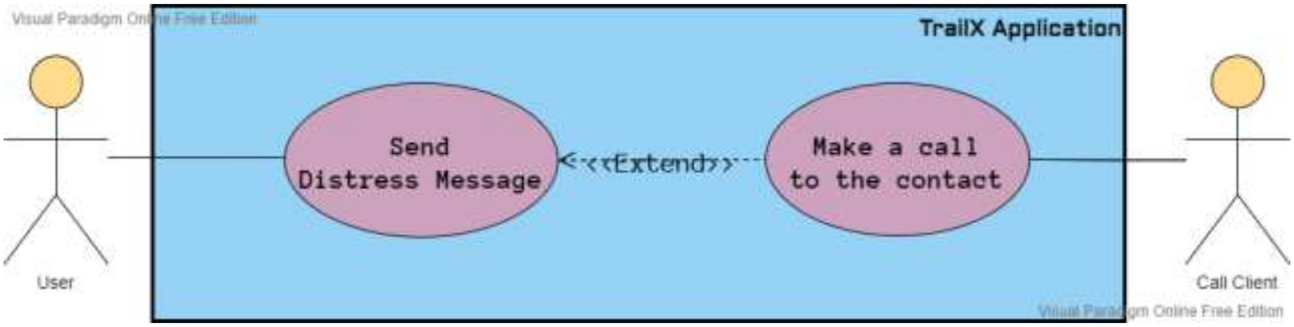
5. Home Screen

Figure 23: Home Screen – Use Case Diagram


Visual Paradigm Online Free Edition
TrailX Application
User
Access My Profile
<<Include>>
Redirect to My Profile Page
Discover New Trails
<<Include>>
Redirect to Discover New trails page
Open Settings & Preferences
<<Include>>
Redirect to Settings page
Access My Trail
<<Include>>
Redirect to My trails page
View Current Trail
<<Include>>
Redirect to active trail page
Emergency SOS
<<Include>>
Redirect to Emergency SOS page
Text
Visual Paradigm Online Free Edition

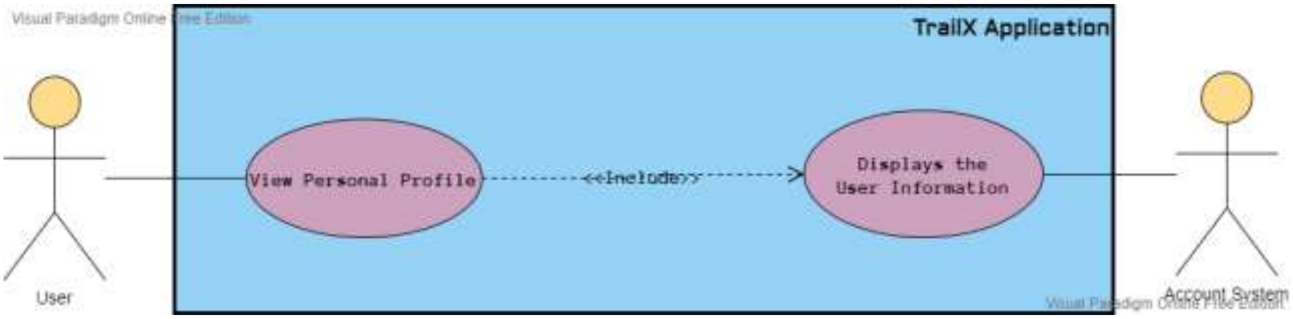
6. Emergency SOS Screen

Figure 24: Emergency SOS Screen – Use Case Diagram



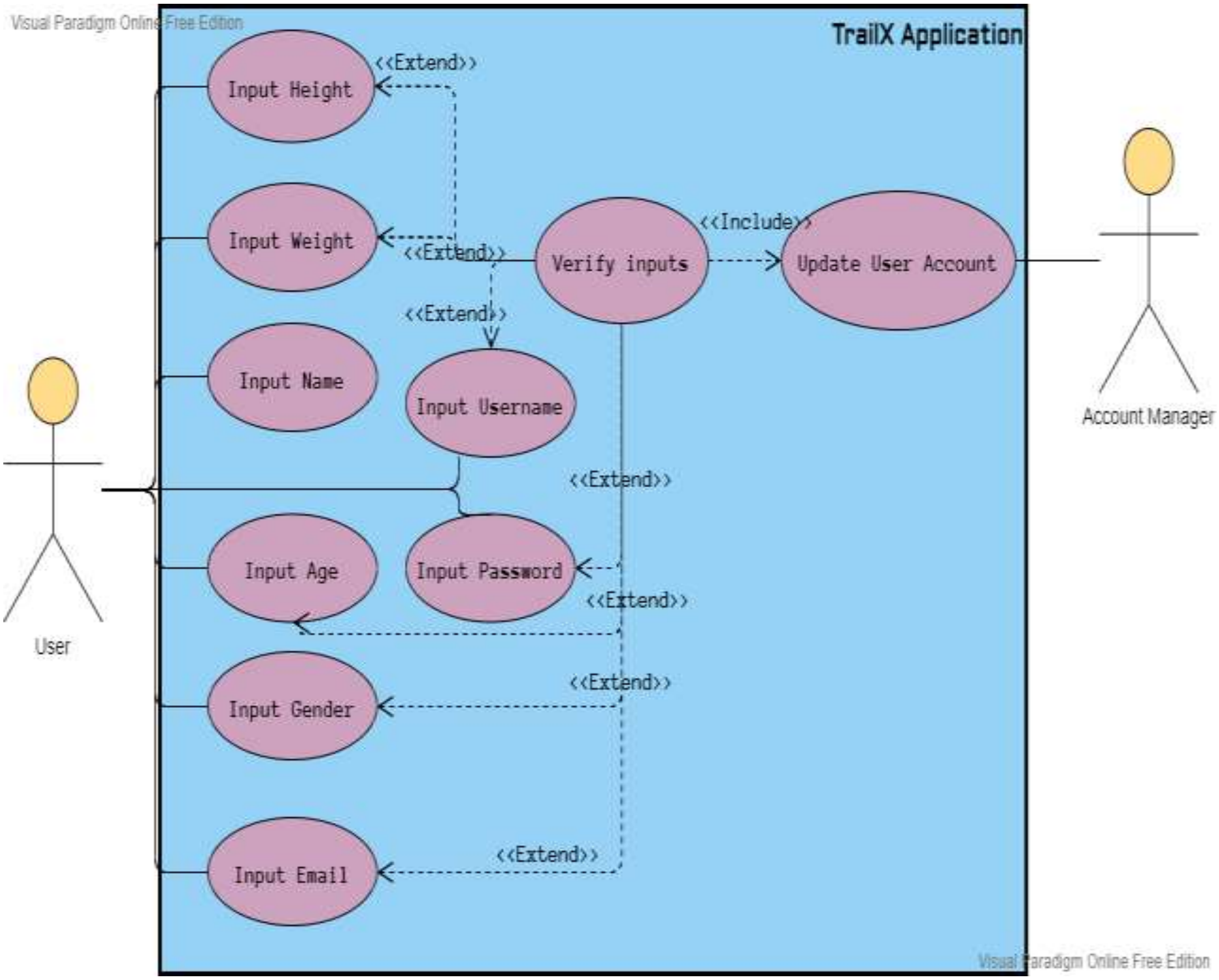
7. My Account Screen

Figure 25: My Account Screen – Use Case Diagram



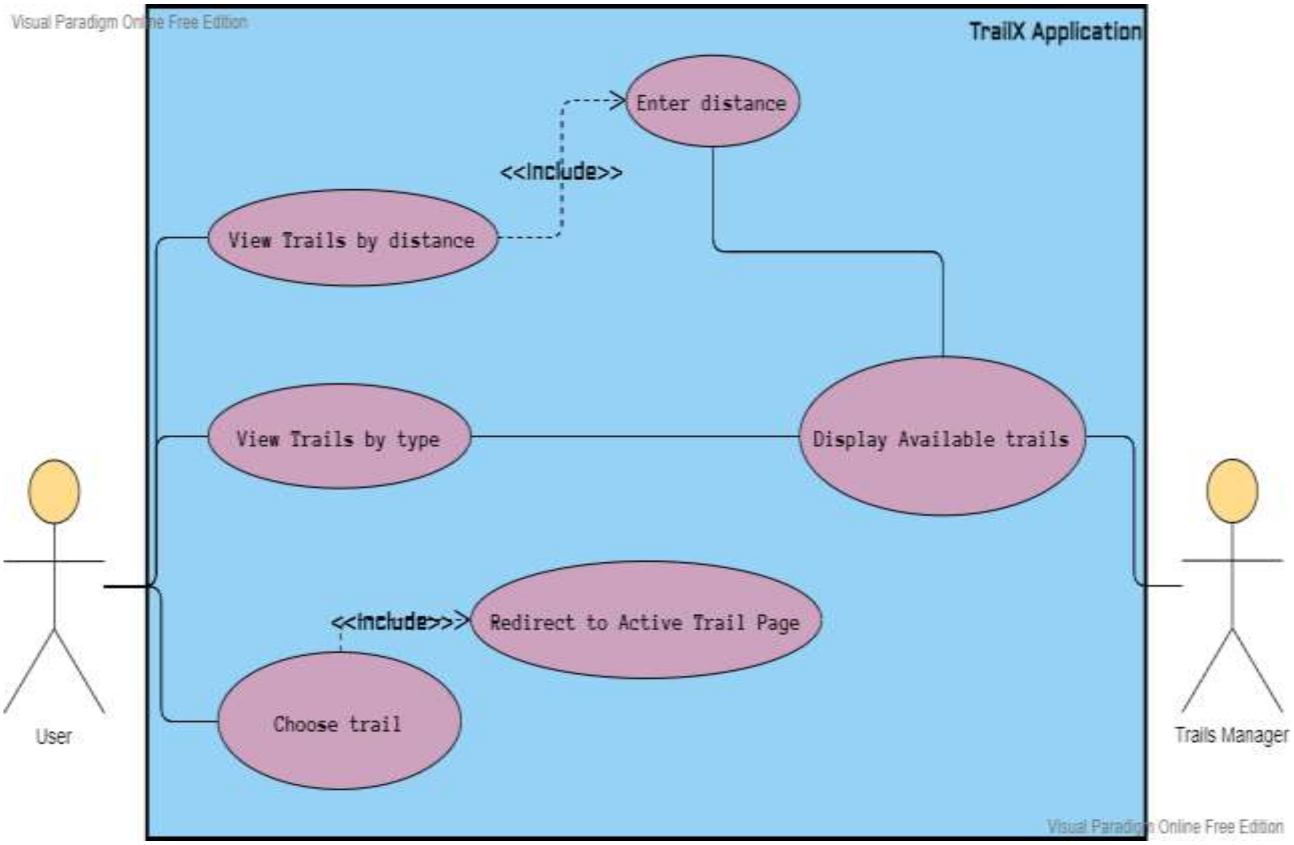
8. Settings Screen

Figure 26: Settings Screen – Use Case Diagram



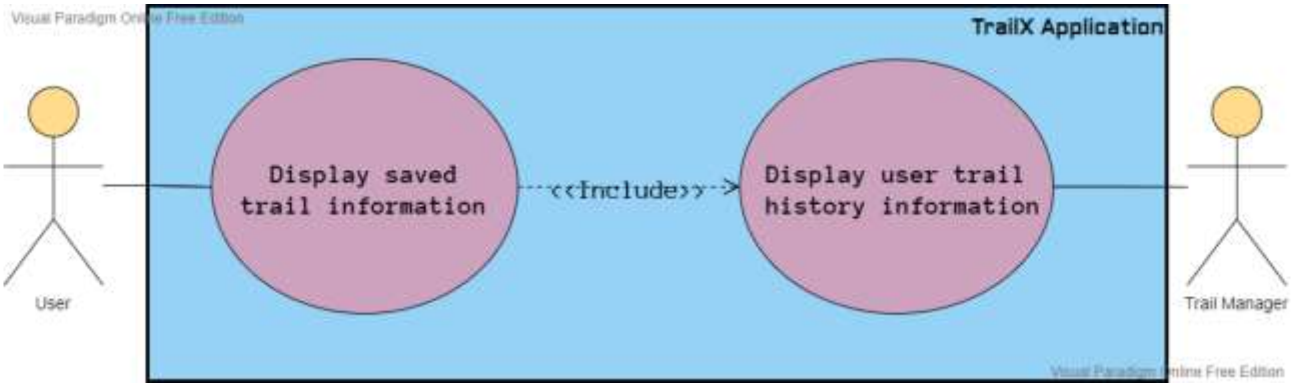
9. Discover New Trails Screen

Figure 27: Discover New Trails Screen – Use Case Diagram



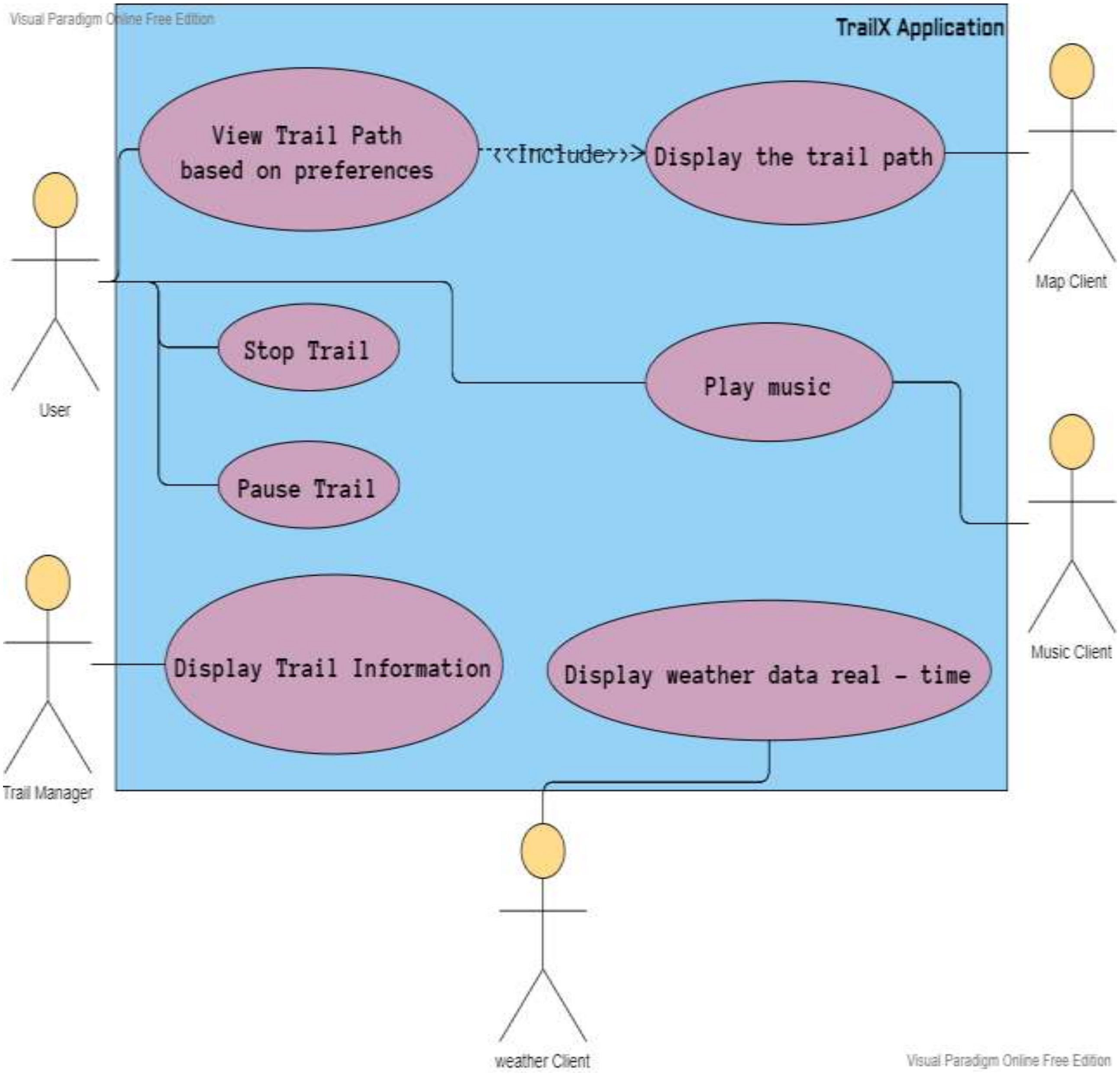
10. My Trails Screen

Figure 28: My Trails Screen – Use Case Diagram


Visual Paradigm Online Free Edition
TrailX Application
Display saved trail information
<<Include>>
Display user trail history information
User
Trail Manager
Visual Paradigm Online Free Edition

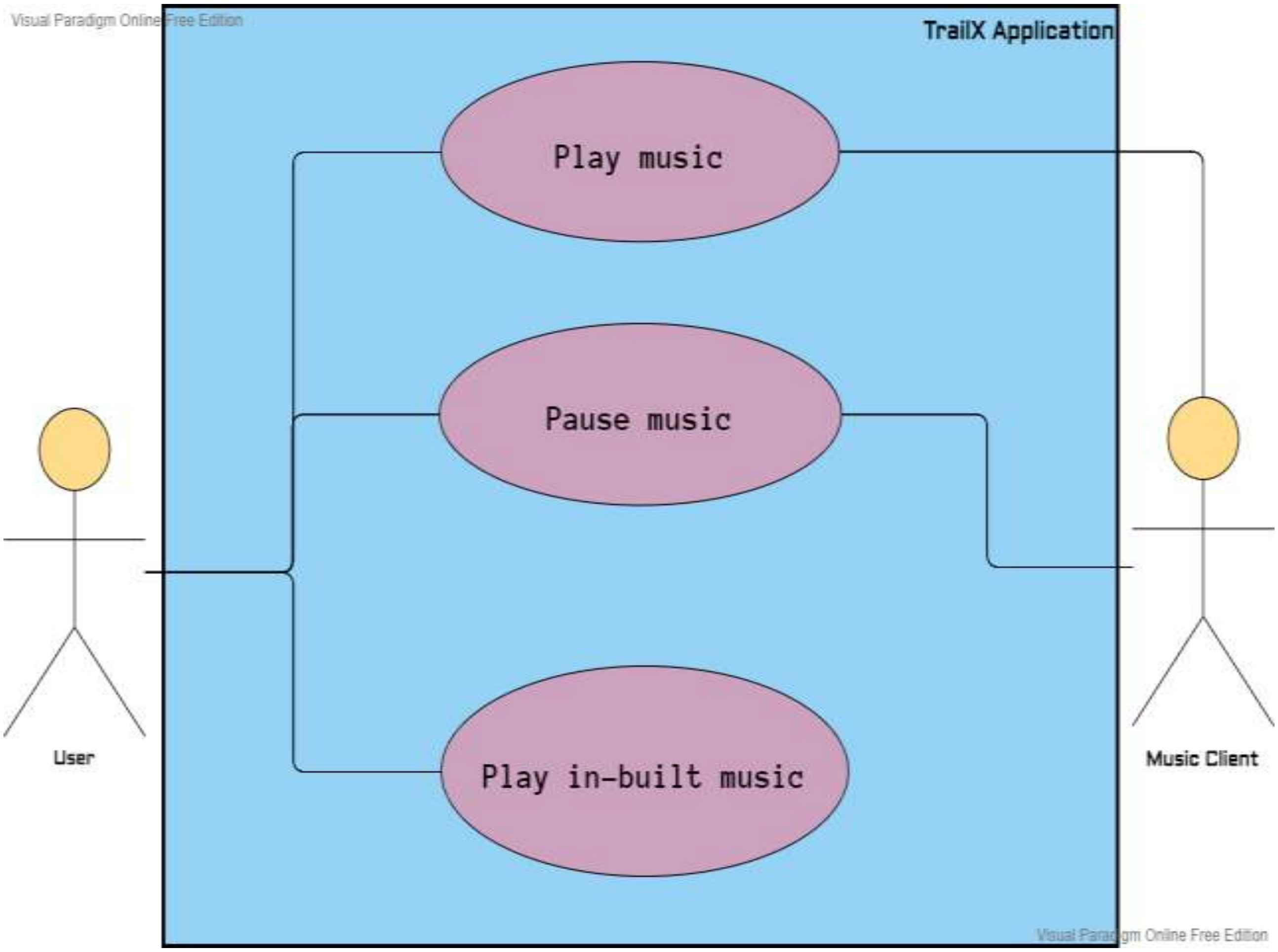
11. Active Trail Screen

Figure 29: Active Trail Screen – Use Case Diagram


Visual Paradigm Online Free Edition
TrailX Application
View Trail Path based on preferences
<<Include>>
Display the trail path
Map Client
User
Stop Trail
Play music
Pause Trail
Music Client
Trail Manager
Display Trail Information
Display weather data real - time
weather Client
Visual Paradigm Online Free Edition

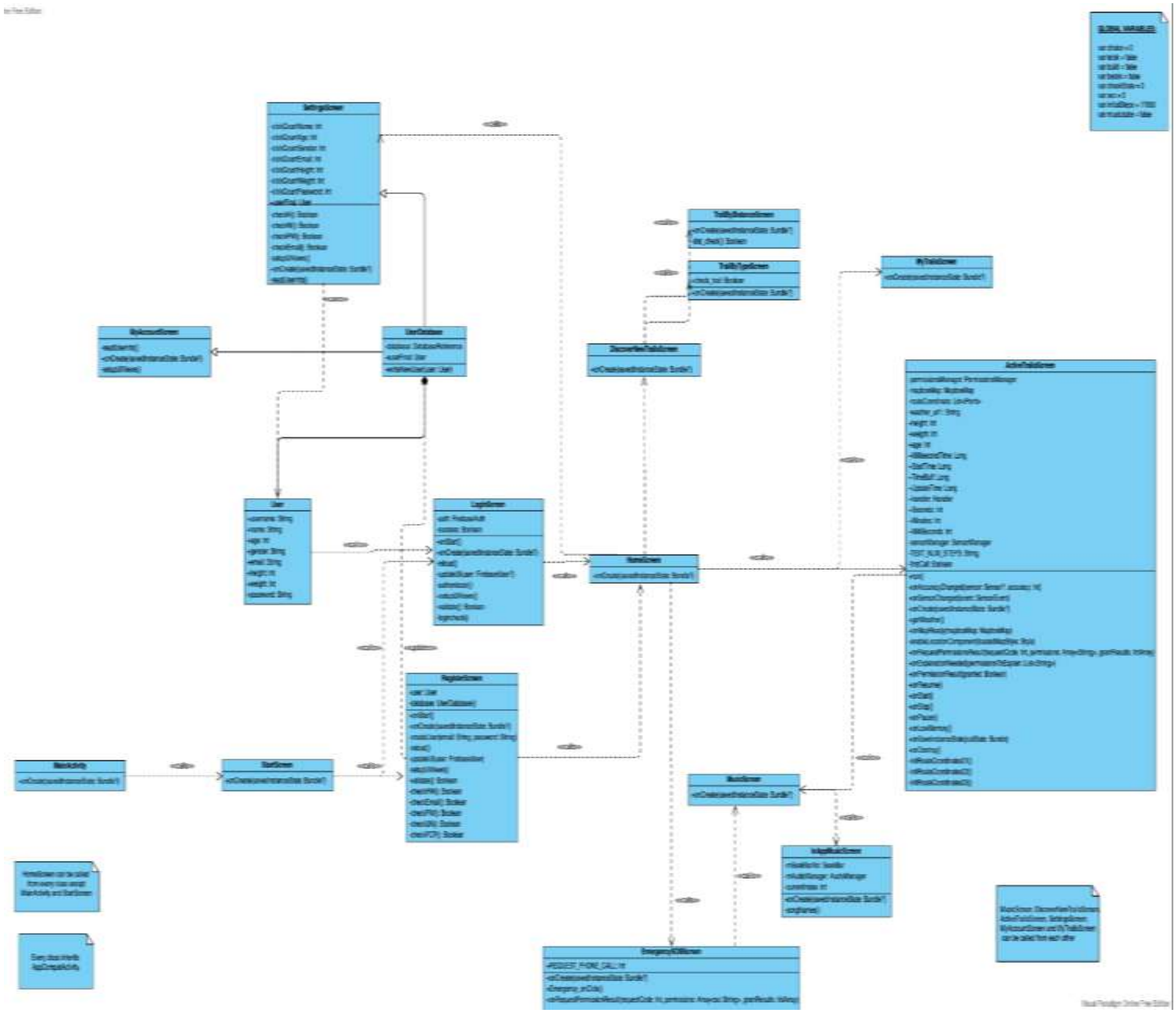
12. Music Screen

Figure 30: Music Screen – Use Case Diagram



Class Diagram

Figure 31: Class Diagram


GLOBAL VARIABLES
SettingsScreen
MyAccountScreen
UserDatabase
User
LoginScreen
HomeScreen
RegisterScreen
MainActivity
StartScreen
TrailsByDistanceScreen
TrailsByTypeScreen
DiscoverNewTrailsScreen
MyTrailsScreen
ActiveTrailsScreen
MusicScreen
EmergencySOSScreen
-REQUEST_PHONE_CALL: int
HomeScreen can be called from every class except MainActivity and StartScreen
Every class inherits AppCompatActivity
MusicScreen, DiscoverNewTrailsScreen, ActiveTrailsScreen, SettingsScreen, MyAccountScreen and MyTrailsScreen can be called from each other
Visual Paradigm Online Free Edition

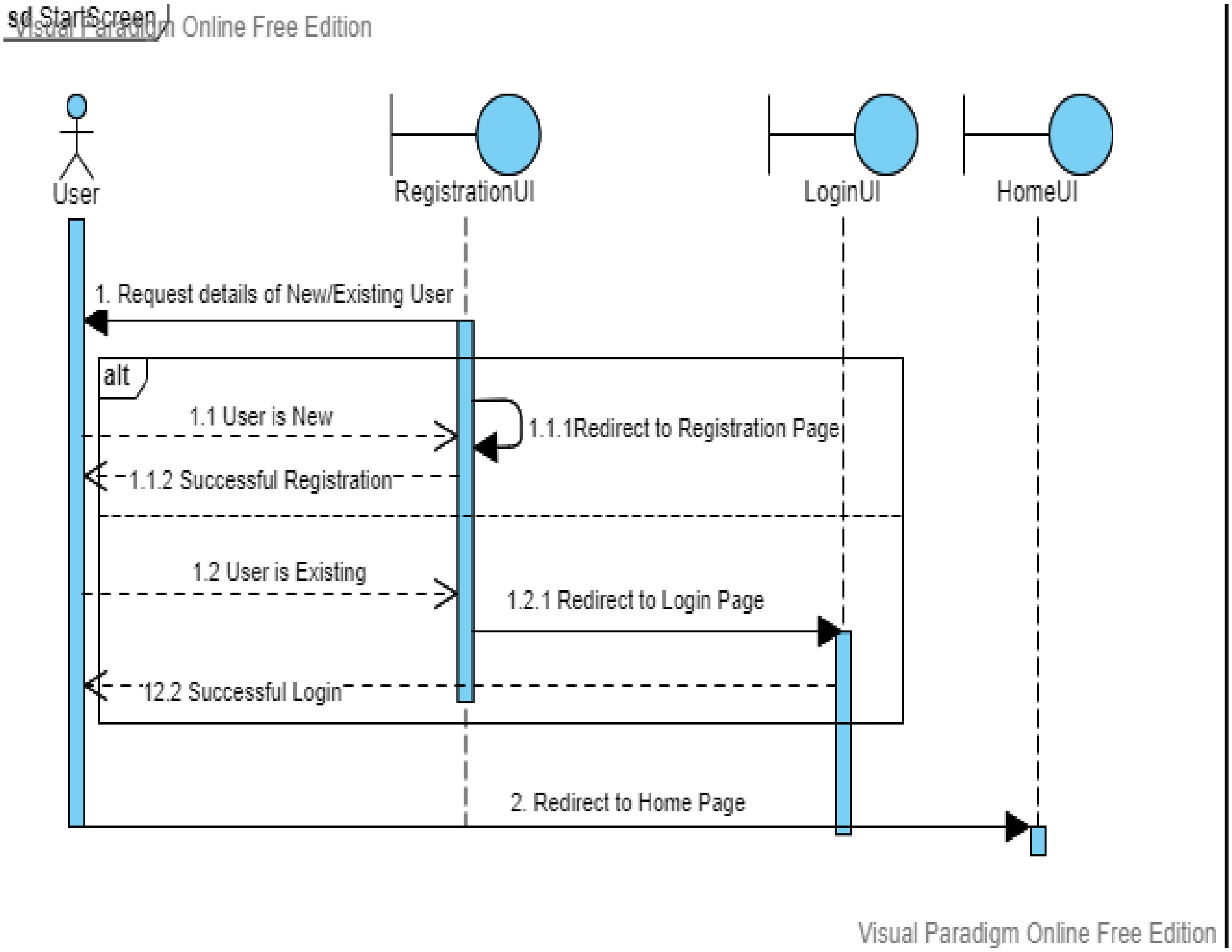
For more clarity:

https://online.visual-paradigm.com/w/xnruqdfo/diagrams/?lightbox=1&highlight=0000ff&edit=https%3A%2F%2Fonline.visual-paradigm.com%2Fw%2Fxnruqdfo%2Fdiagrams%2F%23G1VMoTCq0RK7tgiSb2Qnz31DP_VIAtWr-9&layers=1&nav=1#G1VMoTCq0RK7tgiSb2Qnz31DP_VIAtWr-9

Sequence Diagrams

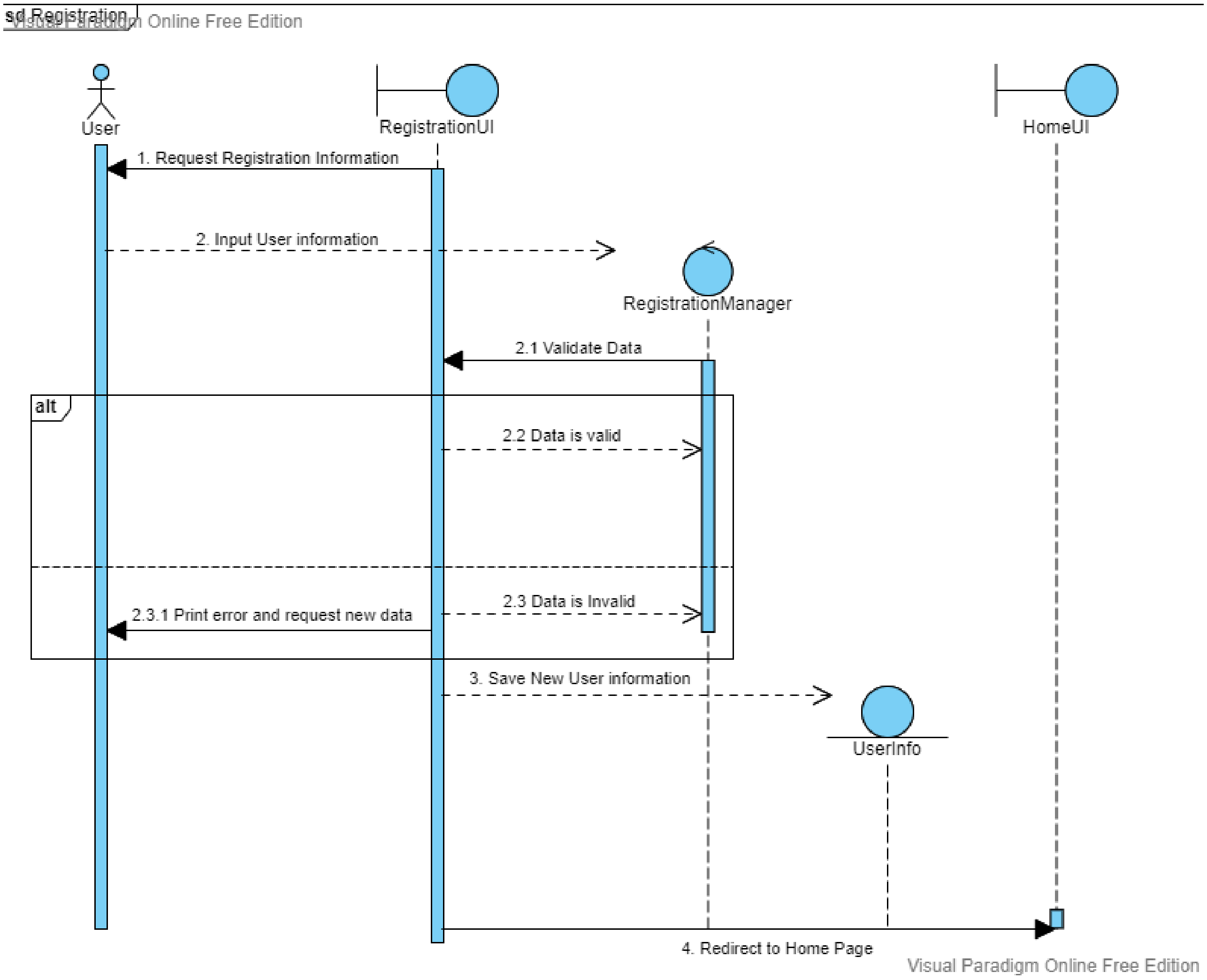
1. Start Screen

Figure 32: Start Screen – Sequence Diagram


sd StartScreen
Visual Paradigm Online Free Edition
User
RegistrationUI
LoginUI
HomeUI
1. Request details of New/Existing User
alt
1.1 User is New
1.1.1Redirect to Registration Page
1.1.2 Successful Registration
1.2 User is Existing
1.2.1 Redirect to Login Page
12.2 Successful Login
2. Redirect to Home Page
Visual Paradigm Online Free Edition

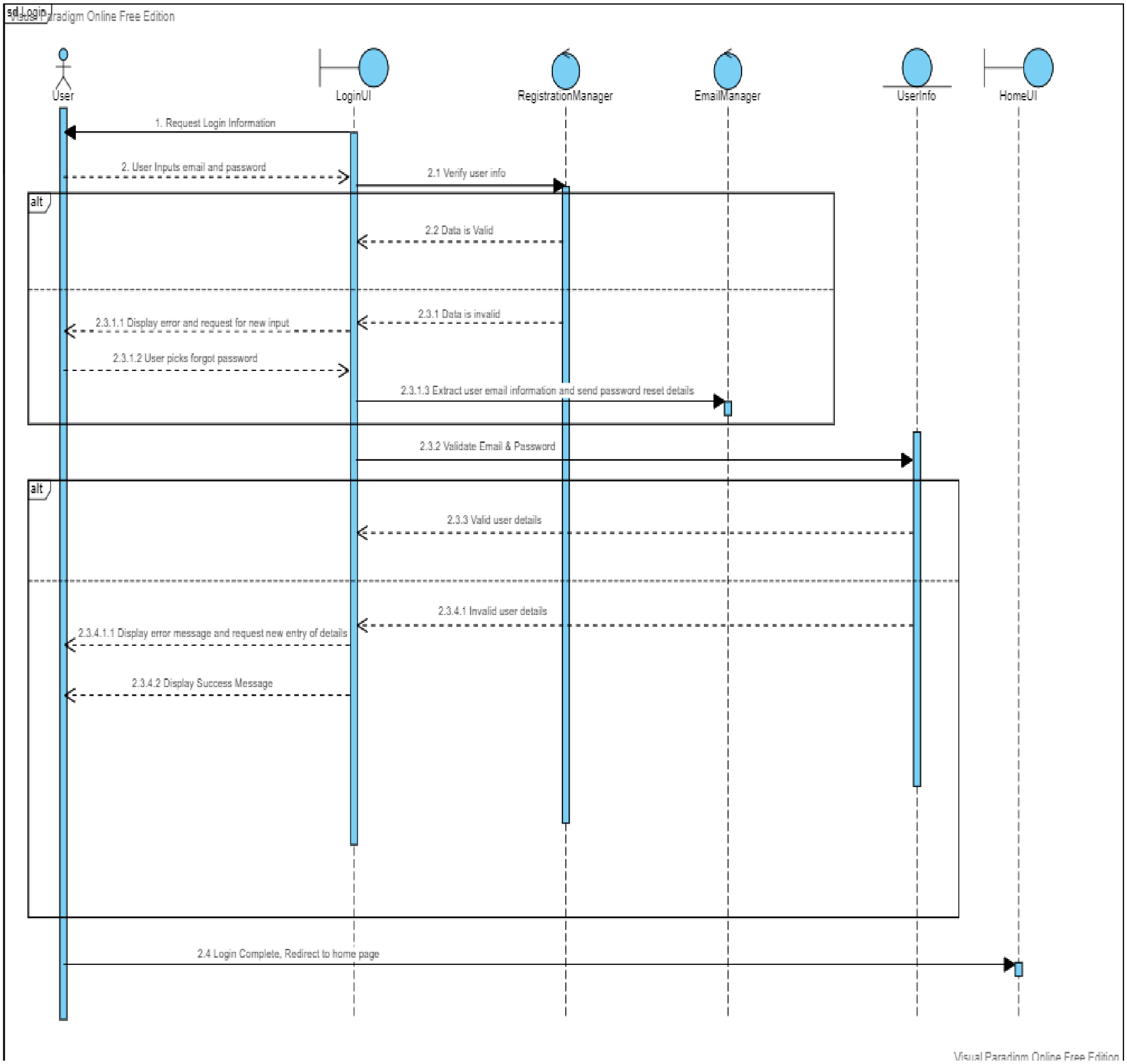
2. Registration Screen

Figure 33: Registration Screen – Sequence Diagram


sd Registration
Visual Paradigm Online Free Edition
User
RegistrationUI
HomeUI
1. Request Registration Information
2. Input User information
RegistrationManager
2.1 Validate Data
alt
2.2 Data is valid
2.3 Data is Invalid
2.3.1 Print error and request new data
3. Save New User information
UserInfo
4. Redirect to Home Page
Visual Paradigm Online Free Edition

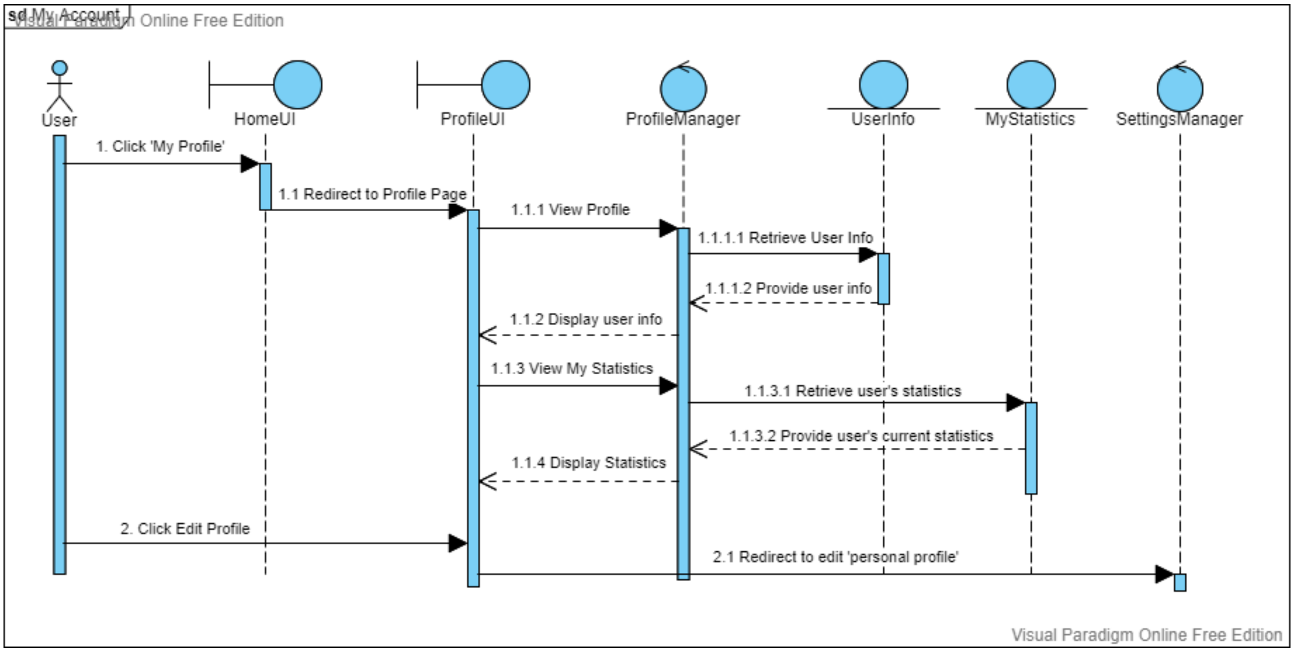
3. Login Screen

Figure 34: Login Screen – Sequence Diagram


sd Login
Visual Paradigm Online Free Edition
User
LoginUI
RegistrationManager
EmailManager
UserInfo
HomeUI
1. Request Login Information
2. User Inputs email and password
2.1 Verify user info
alt
2.2 Data is Valid
2.3.1 Data is invalid
2.3.1.1 Display error and request for new input
2.3.1.2 User picks forgot password
2.3.1.3 Extract user email information and send password reset details
2.3.2 Validate Email & Password
alt
2.3.3 Valid user details
2.3.4.1 Invalid user details
2.3.4.1.1 Display error message and request new entry of details
2.3.4.2 Display Success Message
2.4 Login Complete, Redirect to home page
Visual Paradigm Online Free Edition

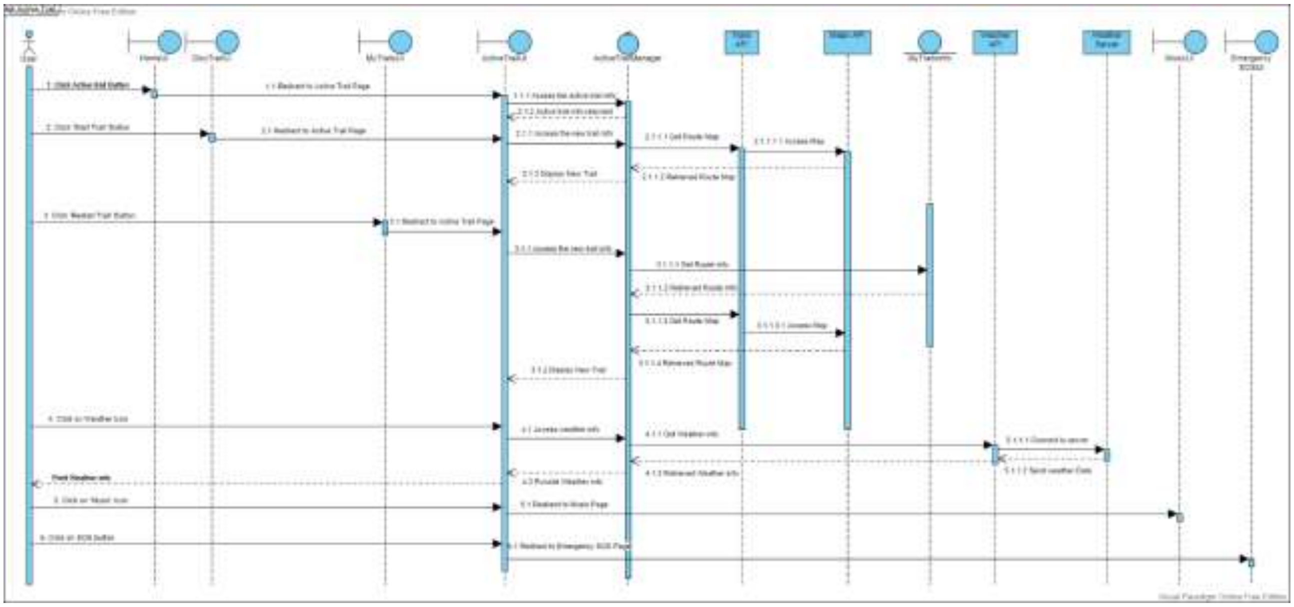
4. My Account Screen

Figure 35: My Account Screen – Sequence Diagram


sd My Account
Visual Paradigm Online Free Edition
User
HomeUI
ProfileUI
ProfileManager
UserInfo
MyStatistics
SettingsManager
1. Click 'My Profile'
1.1 Redirect to Profile Page
1.1.1 View Profile
1.1.1.1 Retrieve User Info
1.1.1.2 Provide user info
1.1.2 Display user info
1.1.3 View My Statistics
1.1.3.1 Retrieve user's statistics
1.1.3.2 Provide user's current statistics
1.1.4 Display Statistics
2. Click Edit Profile
2.1 Redirect to edit 'personal profile'
Visual Paradigm Online Free Edition

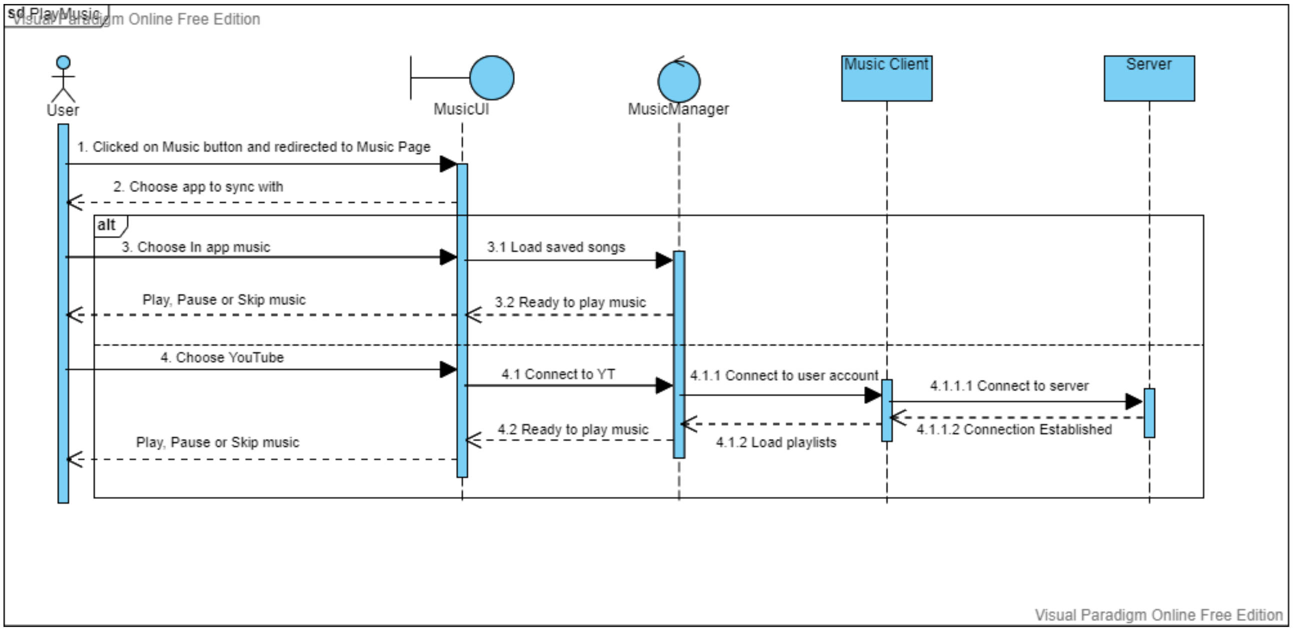
5. Active Trail Screen

Figure 36: Active Trail Screen – Sequence Diagram


User
ActiveTrailManager
1.1 Redirect to Active Trail Page
2.1 Redirect to Active Trail Page
2.1.3 Display New Trail
2.1.1.2 Retrieved Route Map
3.1 Redirect to Active Trail Page
3.1.2 Display New Trail
3.1.1.4 Retrieved Route Map
4.1.1 Get Weather info
4.1.2 Retrieved Weather info
5.1 Redirect to Music Page
6.1 Redirect to Emergency SOS Page

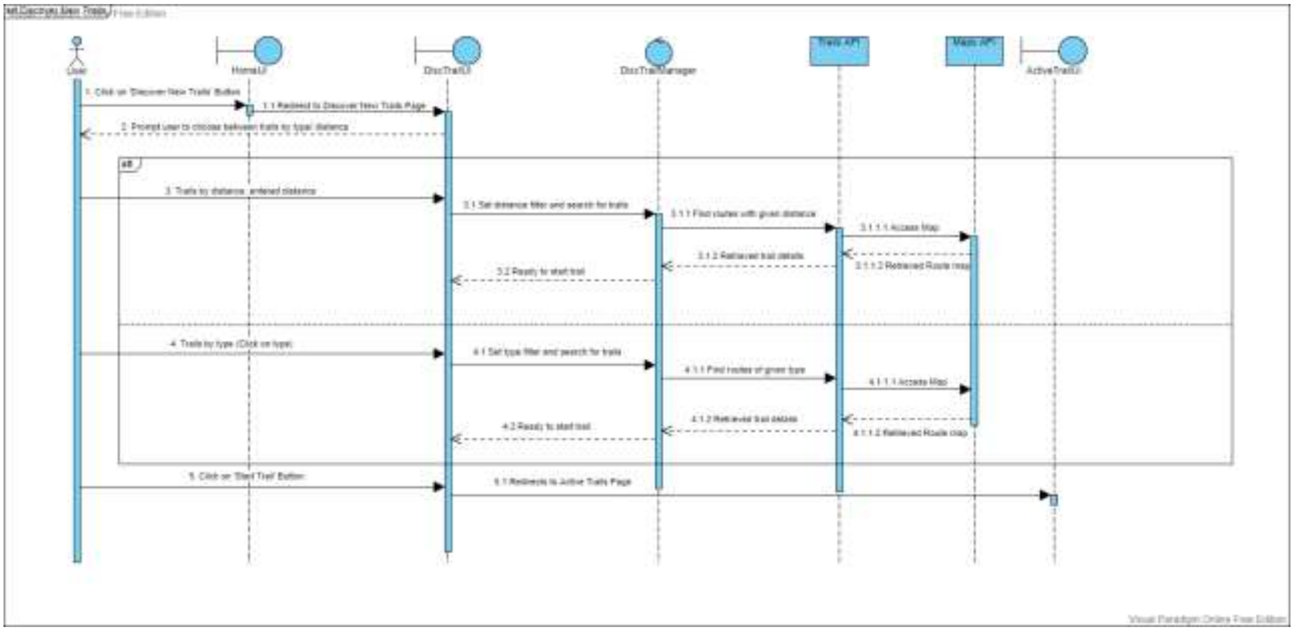
6. Music Screen

Figure 37: Music Screen – Sequence Diagram


sd PlayMusic
Visual Paradigm Online Free Edition
User
MusicUI
MusicManager
Music Client
Server
1. Clicked on Music button and redirected to Music Page
2. Choose app to sync with
alt
3. Choose In app music
3.1 Load saved songs
Play, Pause or Skip music
3.2 Ready to play music
4. Choose YouTube
4.1 Connect to YT
4.1.1 Connect to user account
4.1.1.1 Connect to server
4.2 Ready to play music
4.1.2 Load playlists
4.1.1.2 Connection Established
Play, Pause or Skip music
Visual Paradigm Online Free Edition

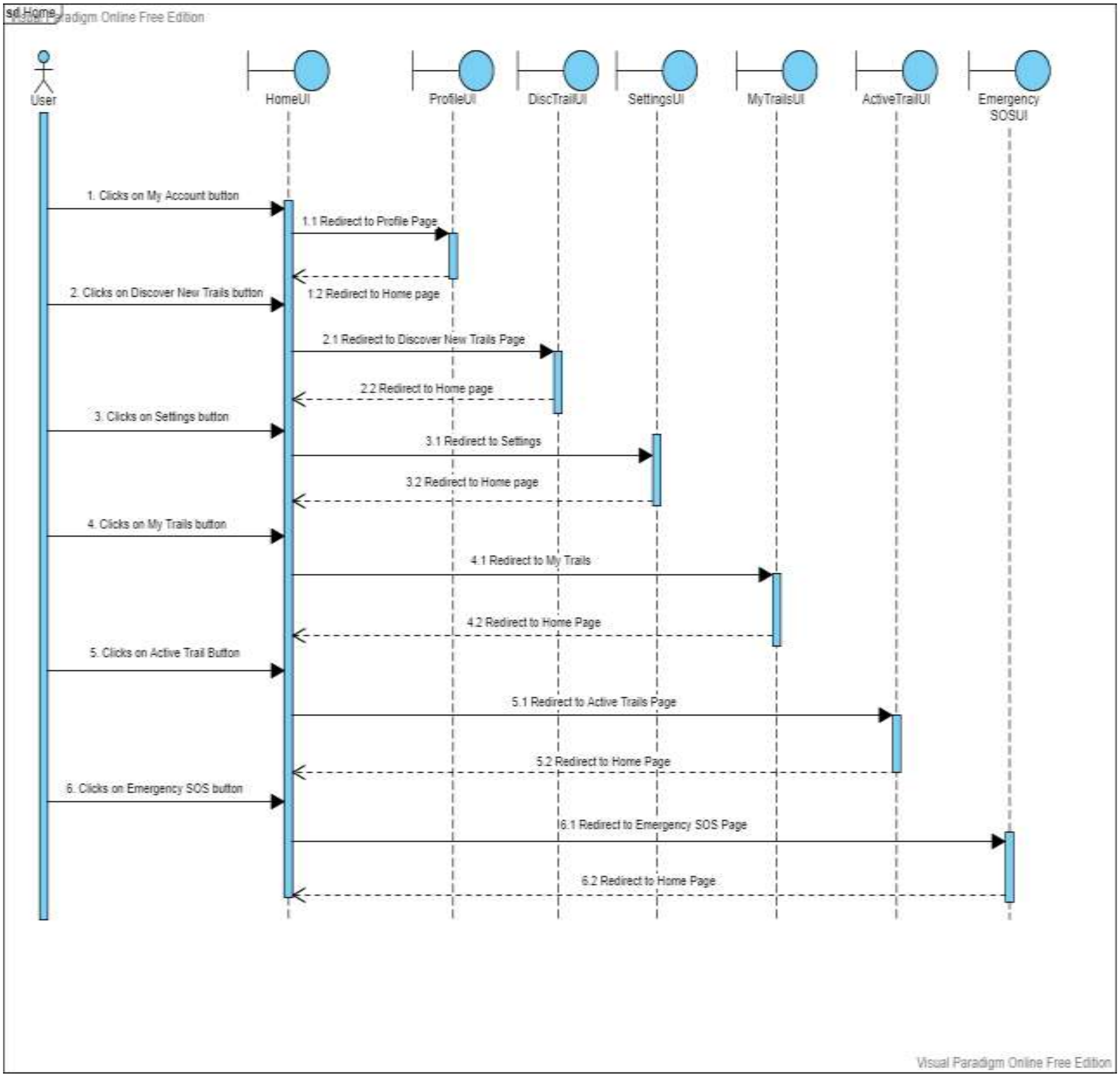
7. Discover New Trails Screen

Figure 38: Discover New Trails Screen – Sequence Diagram


sd Discover New Trails
Free Edition
User
HomeUI
DiscTrailUI
DiscTrailManager
Trails API
Maps API
ActiveTrailUI
1. Click on 'Discover New Trails' Button
1.1 Redirect to Discover New Trails Page
2. Prompt user to choose between trails by type/ distance
alt
3. Trails by distance: entered distance
3.1 Set distance filter and search for trails
3.1.1 Find clusters with given distance
3.1.1.1 Access Map
3.1.2 Retrieved trail details
3.1.1.2 Retrieved Route map
3.2 Ready to start trail
4. Trails by type (Click on type)
4.1 Set type filter and search for trails
4.1.1 Find routes of given type
4.1.1.1 Access Map
4.1.2 Retrieved trail details
4.1.1.2 Retrieved Route map
4.2 Ready to start trail
5. Click on 'Start Trail' Button
5.1 Redirects to Active Trails Page
Visual Paradigm Online Free Edition

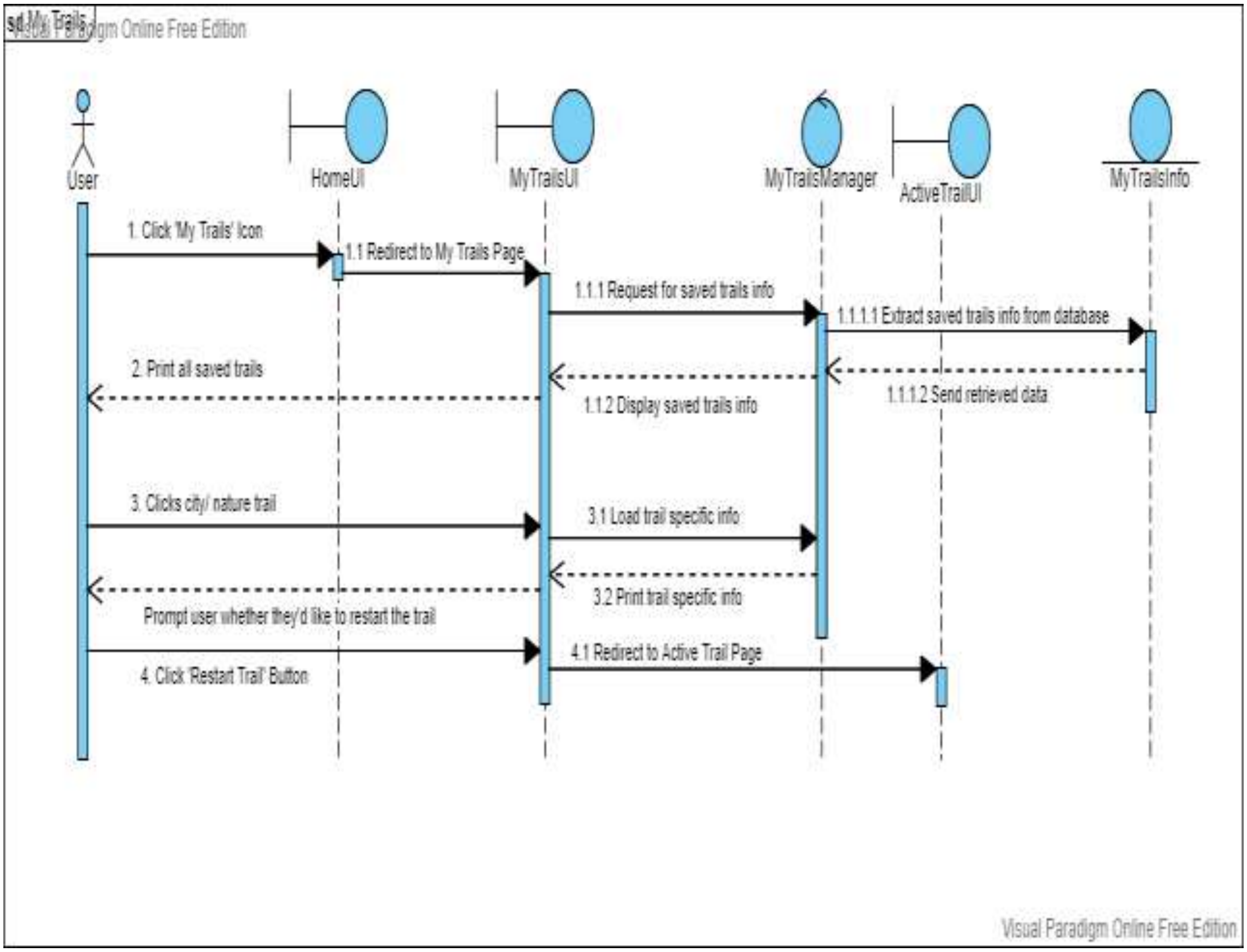
8. Home Screen

Figure 39: Home Screen – Sequence Diagram


sd Home
Visual Paradigm Online Free Edition
User
HomeUI
ProfileUI
DiscTrailUI
SettingsUI
MyTrailsUI
ActiveTrailUI
Emergency SOSUI
1. Clicks on My Account button
1.1 Redirect to Profile Page
1.2 Redirect to Home page
2. Clicks on Discover New Trails button
2.1 Redirect to Discover New Trails Page
2.2 Redirect to Home page
3. Clicks on Settings button
3.1 Redirect to Settings
3.2 Redirect to Home page
4. Clicks on My Trails button
4.1 Redirect to My Trails
4.2 Redirect to Home Page
5. Clicks on Active Trail Button
5.1 Redirect to Active Trails Page
5.2 Redirect to Home Page
6. Clicks on Emergency SOS button
6.1 Redirect to Emergency SOS Page
6.2 Redirect to Home Page
Visual Paradigm Online Free Edition

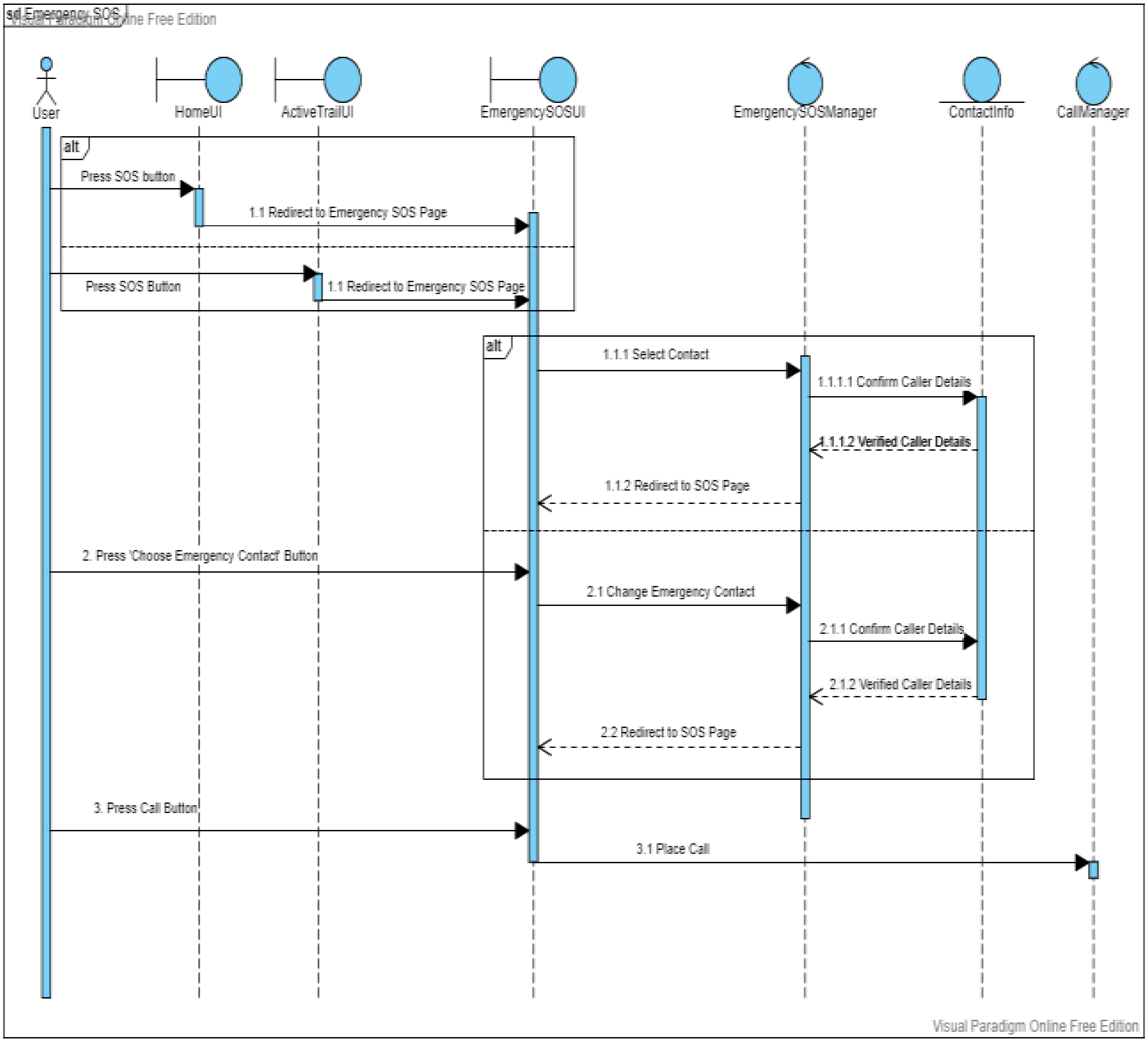
9. My Trails Screen

Figure 40: My Trails Screen – Sequence Diagram


sd My Trails
Visual Paradigm Online Free Edition
User
HomeUI
MyTrailsUI
MyTrailsManager
ActiveTrailUI
MyTrailsInfo
1. Click 'My Trails' Icon
1.1 Redirect to My Trails Page
1.1.1 Request for saved trails info
1.1.1.1 Extract saved trails info from database
2. Print all saved trails
1.1.2 Display saved trails info
1.1.1.2 Send retrieved data
3. Clicks city/ nature trail
3.1 Load trail specific info
3.2 Print trail specific info
Prompt user whether they'd like to restart the trail
4. Click 'Restart Trail' Button
4.1 Redirect to Active Trail Page
Visual Paradigm Online Free Edition

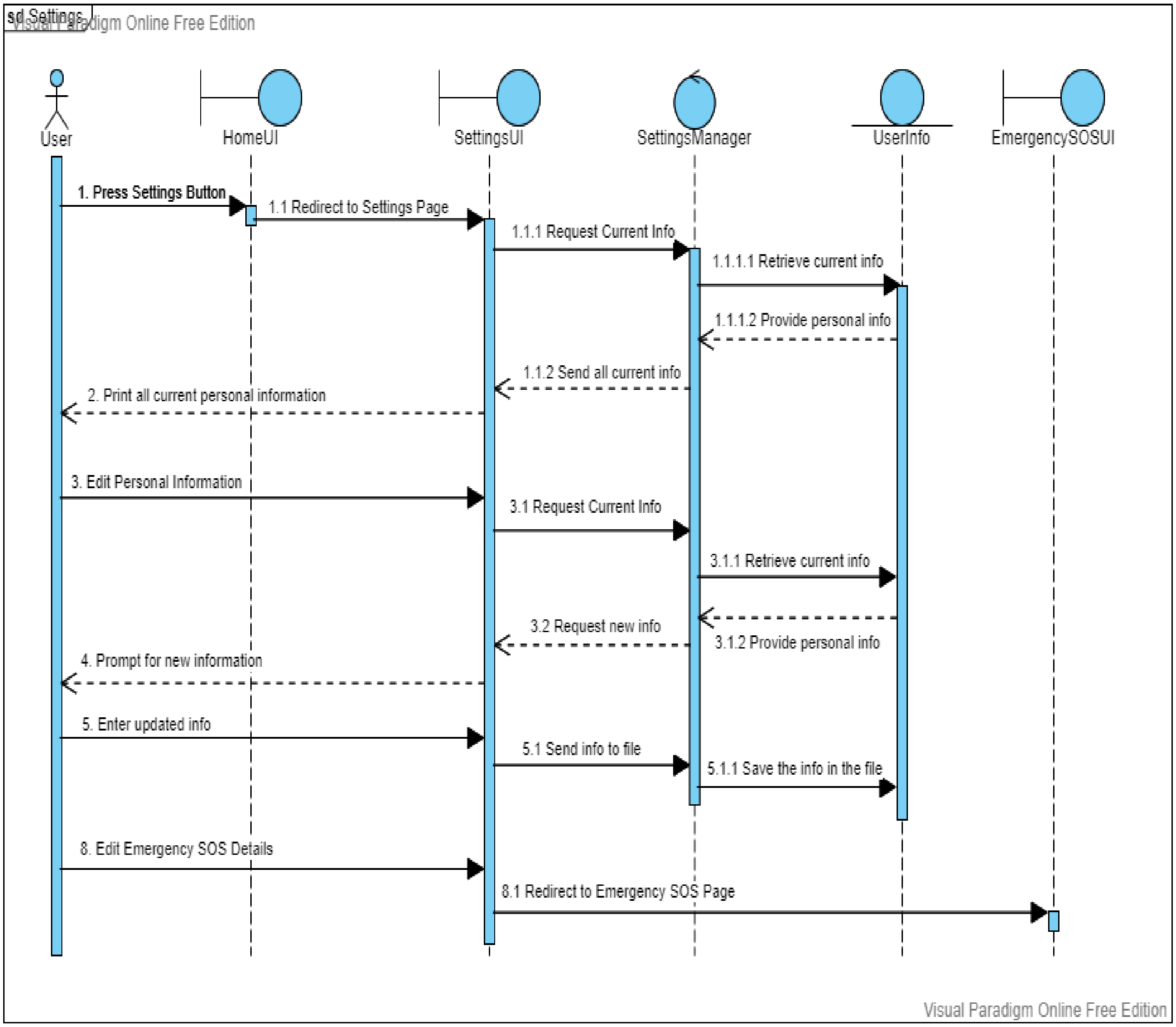
10. Emergency SOS Screen

Figure 41: Emergency SOS Screen – Sequence Diagram


sd Emergency SOS
Visual Paradigm Online Free Edition
User
HomeUI
ActiveTrailUI
EmergencySOSUI
EmergencySOSManager
ContactInfo
CallManager
alt
Press SOS button
1.1 Redirect to Emergency SOS Page
Press SOS Button
1.1 Redirect to Emergency SOS Page
alt
1.1.1 Select Contact
1.1.1.1 Confirm Caller Details
1.1.1.2 Verified Caller Details
1.1.2 Redirect to SOS Page
2. Press 'Choose Emergency Contact' Button
2.1 Change Emergency Contact
2.1.1 Confirm Caller Details
2.1.2 Verified Caller Details
2.2 Redirect to SOS Page
3. Press Call Button
3.1 Place Call
Visual Paradigm Online Free Edition

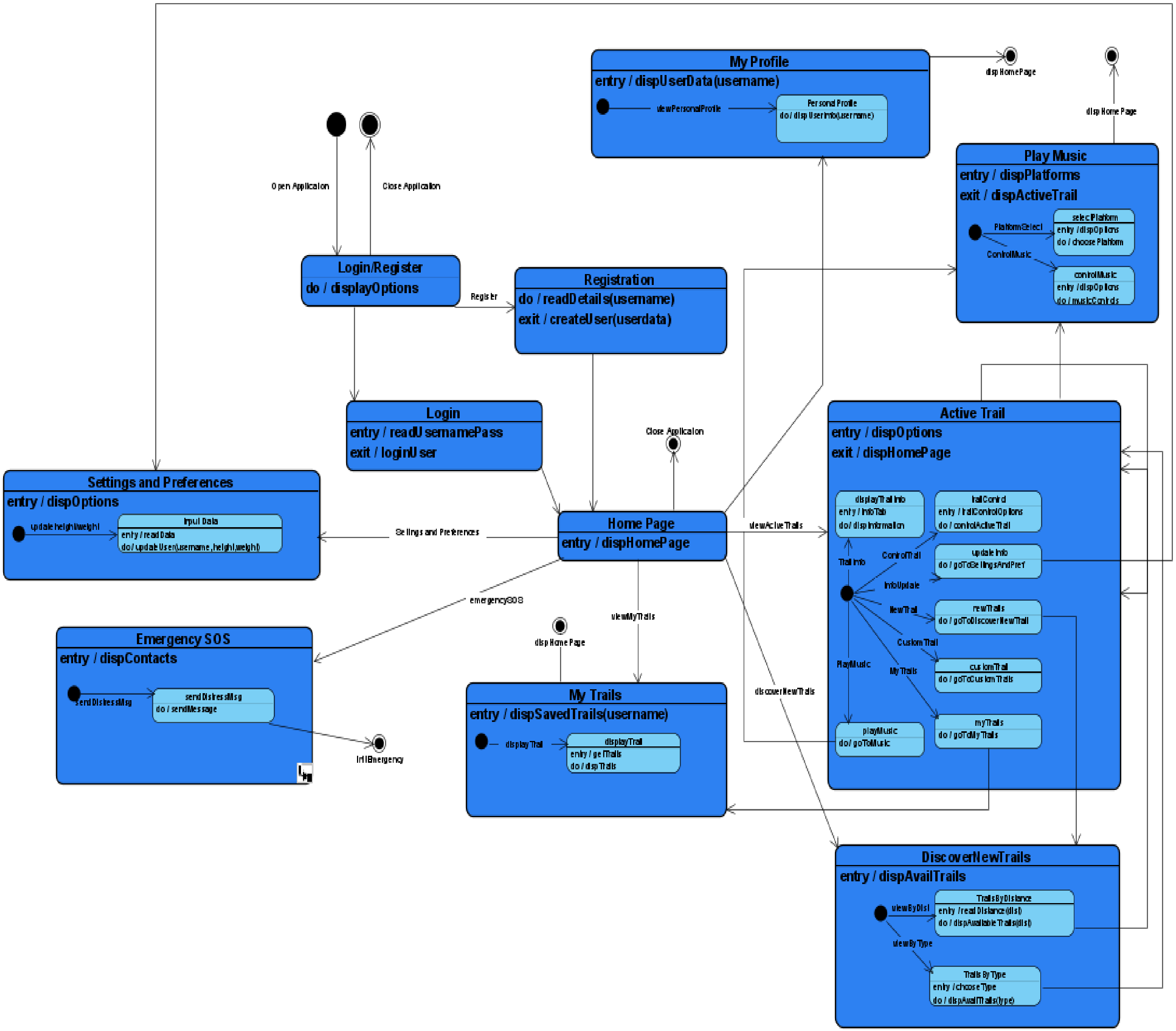
11. Settings Screen

Figure 42: Settings Screen – Sequence Diagram


sd Settings
Visual Paradigm Online Free Edition
User
HomeUI
SettingsUI
SettingsManager
UserInfo
EmergencySOSUI
1. Press Settings Button
1.1 Redirect to Settings Page
1.1.1 Request Current Info
1.1.1.1 Retrieve current info
1.1.1.2 Provide personal info
1.1.2 Send all current info
2. Print all current personal information
3. Edit Personal Information
3.1 Request Current Info
3.1.1 Retrieve current info
3.2 Request new info
3.1.2 Provide personal info
4. Prompt for new information
5. Enter updated info
5.1 Send info to file
5.1.1 Save the info in the file
8. Edit Emergency SOS Details
8.1 Redirect to Emergency SOS Page
Visual Paradigm Online Free Edition

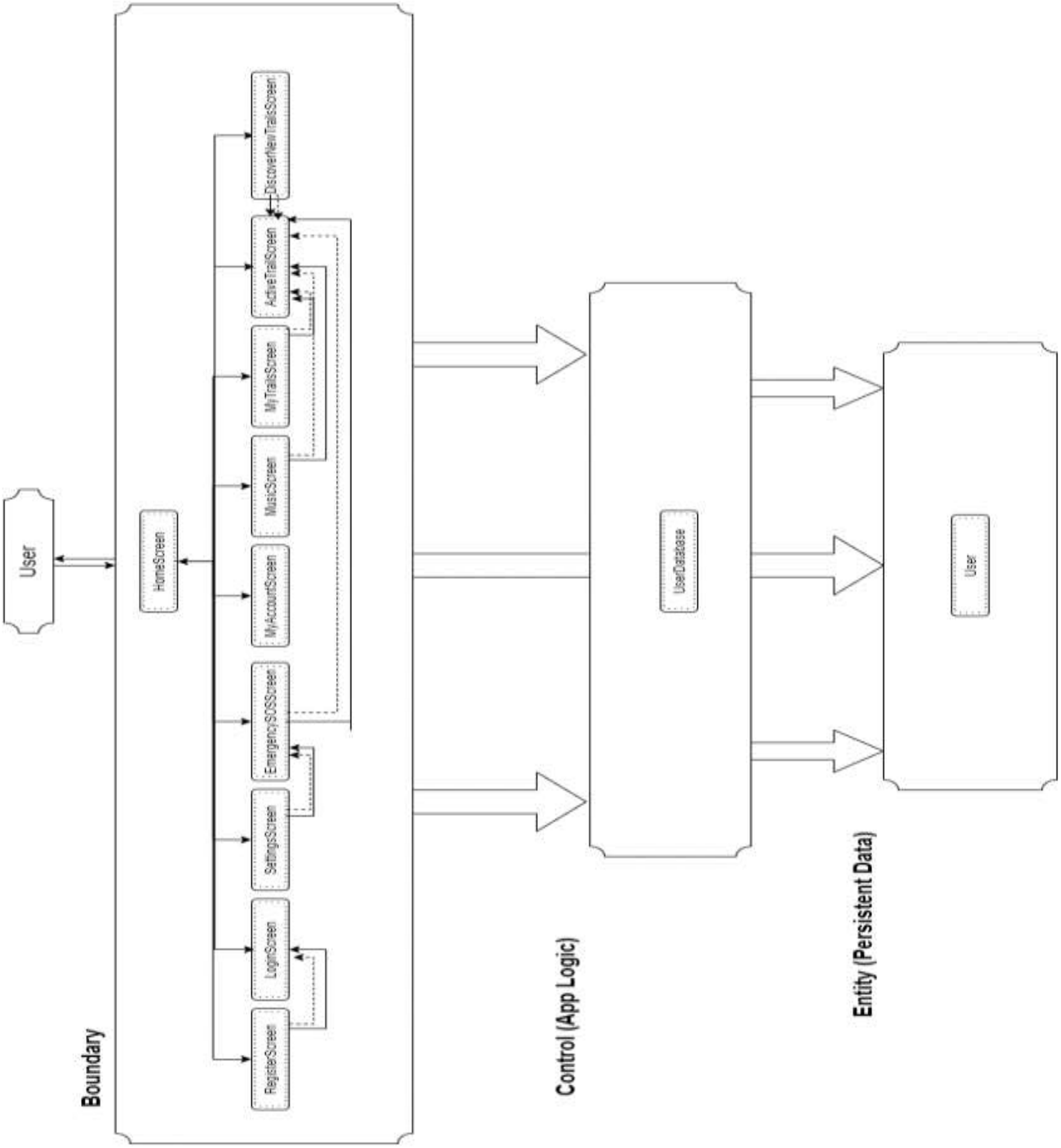
Dialog Map

Figure 43: Application Dialog Map



System Architecture Diagram

Figure 44: System Architecture Diagram



Appendix C: References

Source: http://www.frontiernet.net/~kwiegers/process_assets/srs_template.doc

[1]"How to start to work with Openweather API - OpenWeatherMap", Openweathermap.org, 2021. [Online]. Available: https://openweathermap.org/appid. [Accessed: 03- Apr- 2021].

[2]"Android Basics in Kotlin course | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/courses/android-basics-kotlin/course. [Accessed: 28- Feb- 2021].

[3]"Inline complex XML resources | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/topics/resources/complex-xml-resources. [Accessed: 28- Feb- 2021].

[4]"Show a user's location | Maps SDK | Android", Mapbox, 2021. [Online]. Available: https://docs.mapbox.com/android/maps/examples/show-a-users-location/. [Accessed: 30- Mar- 2021].

[5]Canva, 2021. [Online]. Available: https://www.canva.com/. [Accessed: 13- Apr- 2021].

[6]"Resource types of overview | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/topics/resources/available-resources. [Accessed: 17- Apr- 2021]

[7]"Animation resources | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/topics/resources/animation-resource. [Accessed: 17- Apr- 2021]

[8]"Menu resource | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/topics/resources/menu-resource. [Accessed: 17- Apr- 2021]

[9]"Font resources | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/topics/resources/font-resource. [Accessed: 17- Apr- 2021]

[10]"App Manifest Overview | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/topics/manifest/manifest-intro. [Accessed: 17- Apr- 2021]

[11]"Introduction to Activities | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/components/activities/intro-activities. [Accessed: 17- Apr- 2021]

[12]"App compatibility in Android | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/app-compatibility. [Accessed: 17- Apr- 2021]

[13]"Navigation | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/navigation. [Accessed: 17- Apr- 2021]

[14]"User Interface & Navigation | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/topics/ui. [Accessed: 17- Apr- 2021]

[15]"Services overview | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/components/services. [Accessed: 17- Apr- 2021]

[16]"Permissions on Android | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/topics/permissions/overview. [Accessed: 17- Apr- 2021]

[17]"Build location-aware apps | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/training/location. [Accessed: 17- Apr- 2021]

[18]"Camera API | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/topics/media/camera. [Accessed: 17- Apr- 2021]

[19]"Sensors | Android Developers", Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/topics/sensors. [Accessed: 17- Apr- 2021]

[20]"Add Firebase to your Android project", Firebase, 2021. [Online]. Available: https://firebase.google.com/docs/android/setup. [Accessed: 17- Apr- 2021].

[21]"Installation & Setup on Android | Firebase Realtime Database", Firebase, 2021. [Online]. Available: https://firebase.google.com/docs/database/android/start. [Accessed: 17- Apr- 2021].

[22]"Firebase Authentication", Firebase, 2021. [Online]. Available: https://firebase.google.com/docs/auth. [Accessed: 17- Apr- 2021].