

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

CZ2006 Software Engineering

Software Requirements Specification for SG SPORTS System

Index Group : SS1

Group Name : Anyhow

Group Members :

Name	Metric Number	Contact Number
Koh Luo Hao (Leader)	U1822537H	83666158
Kim Hyunjung	N1904296E	86174926
Teo Wee Ren	U1820326H	92308241
Sven	U1820068B	93294996
Tan Yi Heng	U1821247A	96170121
Filip Skogh	N1904183C	92449607

Report Version : v1.0

TABLE OF CONTENT

1. INTRODUCTION	3
a. Purpose	3
b. Project Scope	3
c. Users and Stakeholders	3
d. Assumptions and Constraints	3
2. FUNCTIONAL REQUIREMENTS	4
a. Use Case Diagrams	7
b. Use Case Descriptions	8
i. Account creation	8
ii. Login	10
iii. View and Edit Profile	12
iv. Search facility	14
v. View facility	16
vi. Book appointment	18
vii. View and edit appointment	19
ix. Add facility	21
x. Delete facility	22
c. Class Diagram	25
d. Sequence Diagram	26
e. State Machine Diagram	32
f. System Architecture Diagram	32
g. Latest UI	33
NON-FUNCTIONAL REQUIREMENTS	46
DATA DICTIONARY	47

1. INTRODUCTION

a. Purpose

The purpose of this document specifies the software requirements for the android application SG Sports. It explains the purpose of the development of the system and defines the scope of the system including functional and non-functional requirements, UML diagrams, use cases, data dictionary and the latest User Interface (UI) of the application.

b. Project Scope

SG Sports is an android application that helps users locate and book sports facilities for their unique needs. The intended product integrates several services into one application, including directions, booking and reviewing of sports facilities, and personalised user reports. Thus, providing a comprehensive platform for users to conveniently take charge and keep track of their fitness and lifestyle.

c. Users and Stakeholders

The stakeholders involved in this project consist of Users and Admins. The Users will be able to use the application features to find the nearest location of the Sports facilities. Users must create an account and sign in with their email and password. They can manage their own profile, facilities booking appointments and write reviews on the facility. Admins will have additional privileges to manage the sports facilities.

d. Assumptions and Constraints

1. Users must always have internet connectivity when using our application.
2. Users must always enable their Global Positioning System (GPS) when using our application.
3. Users need to login in order to access the features in our application.
4. Only the Admin is able to delete or add facilities in our application.
5. System will perform validations before creating or updating the database.

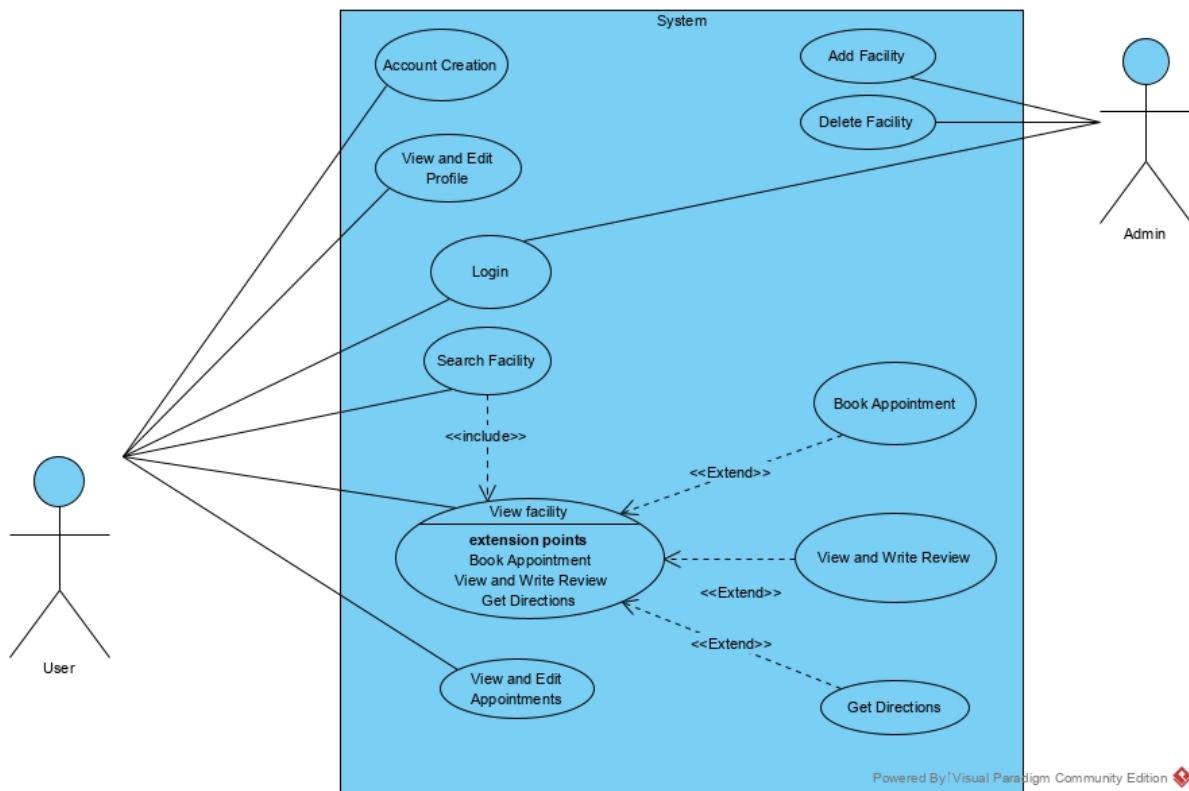
2. FUNCTIONAL REQUIREMENTS

1. Users must create an account to log in to the application.
 - 1.1. Users must choose an email, username and password to create an account.
 - 1.1.1. Password chosen must have a minimum of 6 characters
 - 1.2. The system must validate that the email and username chosen are not already in use.
 - 1.2.1. The application must display an error message if the chosen username and email are already in use.
 - 1.2.2. The error message must prompt the user to choose another username or email
 - 1.3. The system must validate the password to have a minimum of 6 characters.
 - 1.3.1. The application must display an error message if the chosen username or the chosen password don't have a minimum of 6 characters.
2. Users must login to the system to access the application.
 - 2.1. Users must login using the email and password they have chosen during account creation.
 - 2.2. The application must validate that the login credentials are valid.
 - 2.3. The application must display an error message if the login credentials are invalid.
3. Users must be able to manage their profile.
 - 3.1. Users must set a display name.
 - 3.2. Users must have the option of setting their gender.
 - 3.3. Users must have the option of setting their age.
 - 3.4. Users must have the option of setting their mobile number.
4. Users must be allowed to change their account password.
 - 4.1. Users must provide the old password to change their account password.
 - 4.1.1. The application must verify that the set of username and password is stored in the database.

- 4.1.2. The application must display an error message if the set of username and password provided is not found in the database.
 - 4.2. New password chosen must have a minimum of 6 characters.
 - 4.2.1. The application must validate that the new password has a minimum of 6 characters.
 - 4.2.2. The application must display an error message if the new password does not have a minimum of 6 characters.
 - 4.3. User must select ‘confirm to change’ password.
 - 4.3.1. The application must change the old password with the new password in the database.
-
- 5. Users must be able to write reviews for the sporting facilities.
 - 5.1. Users must select one numeric rating between 0,1,2,3,4, and 5.
 - 6. The application must display the average rating of each sporting facility.
 - 6.1. Users must be able to view every rating and comment given by other users.
 - 6.2. The application must display ‘no reviews available’ if a sporting facility does not have any review.
 - 7. The application must allow the users to book appointments at the sporting facility.
 - 7.1. The application must show users a list of available dates(time) for booking.
 - 7.2. Users are restricted to book at most 3 facilities on the list.
 - 7.2.1. The application must cancel the booking process and display an error message if they try to book another facility when they have already booked 3 facilities.
 - 7.2.2. The application must cancel the booking process and display an error message if they try to book the same facility they have booked already.
 - 7.3. The application must display an error message if the booking process fails.

8. Users must be allowed to view the list of appointments they have booked.
 - 8.1. Users must be allowed to cancel any appointments they have booked.
 - 8.2. Users must be allowed to edit any appointments they have booked.
9. The application must return the nearest sporting facility based on the user search requirements.
 - 9.1. Users must enter at least 1 current location before being able to search.
 - 9.1.1. The application must validate that at least 1 current location is provided.
 - 9.1.2. The application must display an error message if no current location is provided.
 - 9.2. Users must be able to search based on the type of sporting facility they require.
 - 9.2.1. The application must allow at most 1 type of sporting facility should users choose to do so.
 - 9.3. Users must be able to search based on the numeric rating of the sporting facilities.
 - 9.3.1. Users can select a minimum rating, up to 1 decimal place. Default value is set to 0.
 - 9.3.2. The application must validate that the minimum rating is within the range of 0-5.
 - 9.3.3. The application must display an error message if the minimum rating is not within the range of 0-5.
10. The application must allow the users to know how to travel to the sporting facility given by the search result.
 - 10.1. The application must allow the users to enter only 1 current location.
 - 10.2. The application must display the mode of travel as well as the estimated time of travel.
11. Admin must be able to manage the list of sporting facilities within the application.
 - 11.1. Admin must be able to add any sporting facility.
 - 11.2. Admin must be able to delete any sporting facility.

a. Use Case Diagrams



b. Use Case Descriptions

i. Account creation

Use Case ID:	1		
Use Case Name:	Account Creation		
Created By:	Yi Heng	Last Updated By:	Yi Heng
Date Created:	2/11/2020	Date Last Updated:	3/14/2020

Actor:	User
Description:	User creates an account.
Preconditions:	Users must have an internet connection. User is not signed in
Postconditions:	The application displays a message stating that the account has been successfully created.
Priority	High
Frequency of Use	Occasionally

Flow of Events:	<ol style="list-style-type: none"> 1. User opens the application. 2. User clicks 'SIGN UP' 3. Application displays account creation form. System requires users to input an email, username, password of minimum 5 characters, confirm password, gender age and mobile number 4. User inputs necessary information correctly.. 5. Application verifies the user's inputs. 6. If the user's input fulfills the requirements, the system proceeds to create an account. 7. Application saves the set of username and password into the application's database. 8. Application displays a message stating the account has been successfully created.
Alternative Flows:	<p>Flow 1: If the password does not match the requirement.</p> <ol style="list-style-type: none"> 1. Application displays a message stating that the password does not fulfil the requirement. 2. Return to step 3 <p>Flow 2: If the password and confirm password are not identical</p> <ol style="list-style-type: none"> 1. Application displays a message stating to confirm password again. 2. Return to step 3 <p>Flow 3: If the email or username exists in the database</p> <ol style="list-style-type: none"> 1. Application displays a message stating that the email or username is already in use. 2. Return to step 3
Exception:	NIL
Includes:	NIL
Extends:	NIL
Assumptions:	NIL

ii. Login

Use Case ID:	2		
Use Case Name:	Login		
Created By:	Yi Heng	Last Updated By:	Yi Heng
Date Created:	14/2/2020	Date Last Updated:	20/3/2020

Actor:	User and Admin
Description:	Users and admin will log in to their account with their email and password at the login page to gain access to the application
Preconditions:	Users and admin must have a registered account.
Postconditions:	Users and admin are granted access to the application after a successful login or denied access after an unsuccessful login
Priority:	High
Frequency of use	Frequently
Flow of Events:	<ol style="list-style-type: none"> 1. Application prompts the user for email and password 2. User enters the email and password and clicks sign in 3. System verifies that the email and password he entered is registered in the database. 4. User successfully login to the system and is redirected to the home page

Alternative Flows:	<p>Flow 1: User inputs invalid email or password</p> <ol style="list-style-type: none">1. System displays a message “Please check your email and password”2. Returns to step 1 <p>Flow 2: User clicks on “SIGN IN WITH GOOGLE”</p> <ol style="list-style-type: none">1. Application displays a prompt allowing the user to indicate which google account to choose.2. User selects a google account3. User successfully login to the system and is redirected to the home page
Exception:	NIL
Includes:	NIL
Extends:	NIL
Assumptions:	The application database contains accurate information of the email and password
Notes and Issues:	NIL

iii. View and Edit Profile

Use Case ID:	3		
Use Case Name:	View and Edit Profile		
Created By:	Yi Heng	Last Updated By:	Yi Heng
Date Created:	11/2/2020	Date Last Updated:	14/3/2020

Actor:	User
Description:	User is able to view and edit his profile details
Preconditions:	User must be able to successfully Login
Postconditions:	User is in profile page and profile details are updated after editing
Priority	High
Frequency of use	Occasionally
Flow of Events:	<ol style="list-style-type: none"> 1. User selects “Profile” from sidebar in homepage 2. System retrieves user profile details from database 3. System displays user profile details and buttons “Edit Profile”, “Cancel” and “Log out”. 4. User clicks on “Edit Profile” 5. System displays the Edit Profile Form and user input data for “Name”, “Age” and “Mobile no.” 6. User clicks Update profile and application verifies the input 7. Application displays a message stating that the profile has been successfully updated.

	8. System displays updated profile page
Alternative Flows:	Flow 1: Data entered in Edit Profile Form is invalid/empty 1. System displays a relevant error message on invalid input 2. System returns to step 5
Exception:	NIL
Includes:	NIL
Extends:	NIL
Assumptions:	User profile exists in database
Notes and Issues:	NIL

iv. Search facility

Use Case ID:	4		
Use Case Name:	Search facility		
Created By:	Yi Heng	Last Updated By:	Yi Heng
Date Created:	11/2/2020	Date Last Updated:	11/3/2020

Actor:	User
Description:	User search for facilities based on his/her requirements
Preconditions:	User must be able to successfully Login
Postconditions:	Displays facilities that meet the user's requirements
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on “Search facilities” and system displays a search bar with “search” and “filter” buttons 2. User types keywords in the search bar and clicks on “filter” 3. System displays 2 options for filtering: “type(s)” and “rating” 4. User selects his/her requirements and clicks “enter” 5. User clicks “search” 6. System will search for and display the ideal facilities based on user's requirements and keywords 7. User clicks on the desired facility and the system will run UseCase 5 : View Facility.

Alternative Flows: <ul style="list-style-type: none"> Flow 1: User keys in keywords and clicks “search” right after <ul style="list-style-type: none"> • System will search for and display the ideal facilities based on the keywords Flow 2: User clicks on the “filter” button, fills in the 2 options without typing anything in the search bar, and clicks “search” <ul style="list-style-type: none"> • System will search for and display the ideal facilities based on the user's requirements Flow 3: System fails to find facilities that satisfy user requirements <ul style="list-style-type: none"> • System will not display any facility • Returns to Step 1
Exception: NIL
Includes: View facility
Extends: Nil
Assumptions: All keywords provided by the user is related to the name of the facility
Notes and Issues: NIL

v. View facility

Use Case ID:	5		
Use Case Name:	View facility		
Created By:	Yi Heng	Last Updated By:	Yi Heng
Date Created:	11/2/2020	Date Last Updated:	17/2/2020

Actor:	User
Description:	User is able to view the facility chosen
Preconditions:	User is login to system and has clicked on a facility marker
Postconditions:	System displays sport facility details and 5 actions:"Info", "Reviews", "Directions", "Book Appointment", and "Write a Review"
Priority	High
Frequency of use	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on facility marker 2. System displays the facility name, address and type of facilities and location on the map.. 3. System displays:"Info", "Reviews", Directions", "Write a review" and "Book Appointment"
Alternative Flows:	<p>Flow 1: User clicks on "Reviews"</p> <ol style="list-style-type: none"> 1. System displays a list of reviews of the facility 2. User clicks one of the reviews 3. System displays the review details page <p>Flow 2: User clicks on "Directions"</p> <ol style="list-style-type: none"> 1. System displays instructions on the best possible way to reach the destination from the current location <p>Flow 3: User clicks on "Write a review"</p> <ol style="list-style-type: none"> 1. System displays write review page 2. User selects the number of stars and is able to write a review that meets the criteria. 3. user clicks submit and returns to the facility details page

	Flow 4: User clicks on “Book Appointment” 1. System will run UseCase 6: Book Appointment
Exception:	NIL
Includes:	NIL
Extends:	View and write review Get directions Book appointment
Assumptions:	Facility details in database are accurate
Notes and Issues:	NIL

vi. Book appointment

Use Case ID:	6		
Use Case Name:	Book Appointment		
Created By:	Wee Ren	Last Updated By:	Wee Ren
Date Created:	11/2/2020	Date Last Updated:	23/2/2020

Actor:	User
Description:	User is able to book appointments through the app
Preconditions:	User is able to login to the system, has clicked on a facility button and is able to choose 'Book Appointment'.
Postconditions:	System displays appointments details and shows appointment is successfully booked message
Priority	High
Frequency of use	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. System displays the available facilities and time slots for booking 2. User selects facility and timeslot and clicks confirm 3. System will verify and store the information in the database 4. System shows appointment is booked
Alternative Flows:	<p>Flow 1: No more slots available</p> <ol style="list-style-type: none"> 1. System informs user that there are no more slots 2. Remain at view facility
Exception:	Nil
Includes:	Nil

Extends:	Nil
Assumptions:	The facility allows booking
Notes and Issues:	Nil

vii. View and edit appointment

Use Case ID:	7		
Use Case Name:	View and edit appointment		
Created By:	Wee Ren	Last Updated By:	Wee Ren
Date Created:	11/2/2020	Date Last Updated:	23/2/2020

Actor:	User
Description:	User is able to view and edit appointments
Preconditions:	User is able to login to the system
Postconditions:	System displays all appointments details and the edit appointment successful message
Priority	High
Frequency of use	Frequent
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on "My reviews" 2. System display all appointments the user has made 3. User chooses an appointment 4. User choose to edit or delete appointment 5. User choose to edit appointment 6. System shows available dates and time slots for user to change 7. User confirm changes and clicks "Update appointment"

Alternative Flows:	Flow 1: User cancel appointment 1. System display all appointments 2. User choose an appointment 3. User choose to delete appointment 4. User finalise and confirm changes
Exception:	NIL
Includes:	NIL
Extends:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

ix. Add facility

Use Case ID:	8		
Use Case Name:	Add Facility		
Created By:	Luo Hao	Last Updated By:	Luo Hao
Date Created:	11/2/2020	Date Last Updated:	23/2/2020

Actor:	Admin
Description:	Admin is able to add a new Facility record.
Preconditions:	Admin is able to login to the system
Postconditions:	System creates a new Facility record.
Priority	Medium
Frequency of use	Occasionally
Flow of Events:	<ol style="list-style-type: none"> 1. Admin selects the action “Add Facility” on the UI. 2. System displays the Add Facility form for the Admin: Name of Facility, Address, Latitude, Longitude, Description, Facility Type. 3. Admin enters the required fields and submits the add facility form. 4. System validates the data entered. 5. System creates a new facility. 6. System displays a successful message.
Alternative Flows:	<p>Flow 1: The data entered by Admin is an invalid or empty field.</p> <ol style="list-style-type: none"> 1. System displays a relevant error message. 2. System returns to step 3.
Exception:	NIL

Includes:	NIL
Extends:	NIL
Assumptions:	NIL
Notes and Issues:	

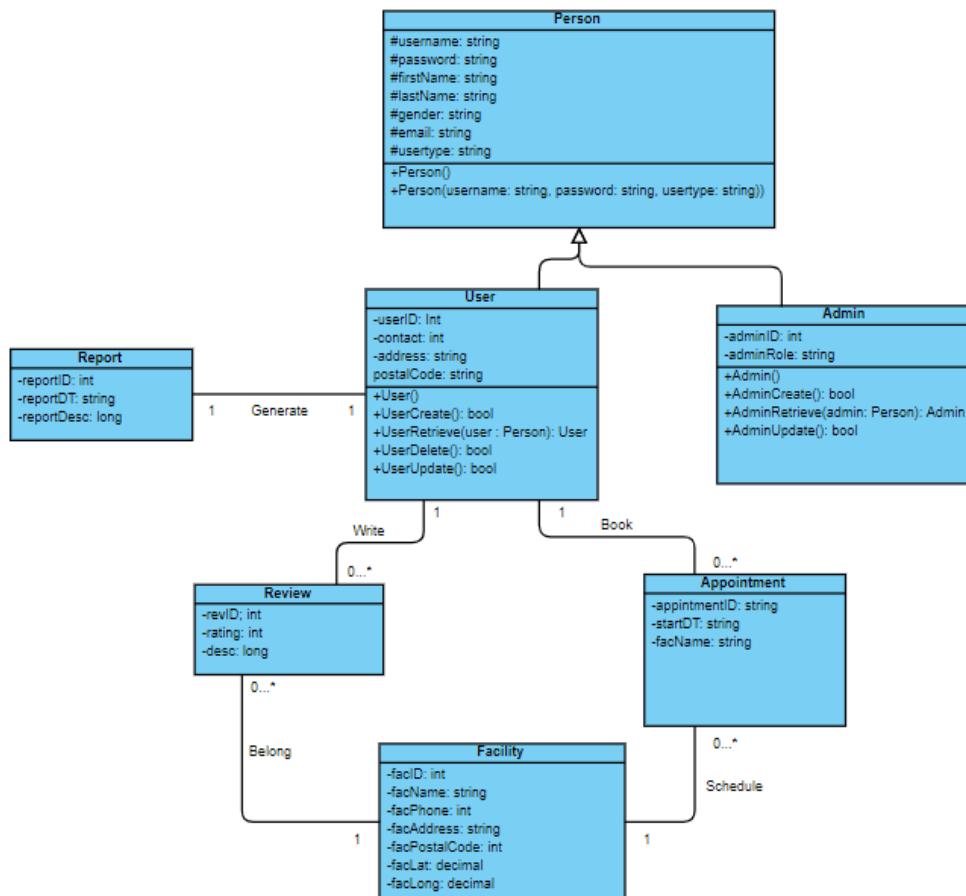
x. Delete facility

Use Case ID:	9		
Use Case Name:	Delete Facility		
Created By:	Luo Hao	Last Updated By:	Luo Hao
Date Created:	11/2/2020	Date Last Updated:	23/2/2020

Actor:	Admin
Description:	Admin is able to delete the Facility record.
Preconditions:	Admin is able to login to the system
Postconditions:	System updates the facility record.
Priority	Low
Frequency of use	Rarely
Flow of Events:	<ol style="list-style-type: none"> 1. Admin selects the action “Delete Facility” on the UI. 2. System shows a list of all facilities 3. Admin selects one of the facility records 4. System displays confirmation window 5. Admin selects ‘Yes’ 6. System removes the facility record 7. System displays a successful message 8. System redirects the user to the List of Current Facilities page.
Alternative Flows:	NIL
Exception:	NIL
Includes:	NIL

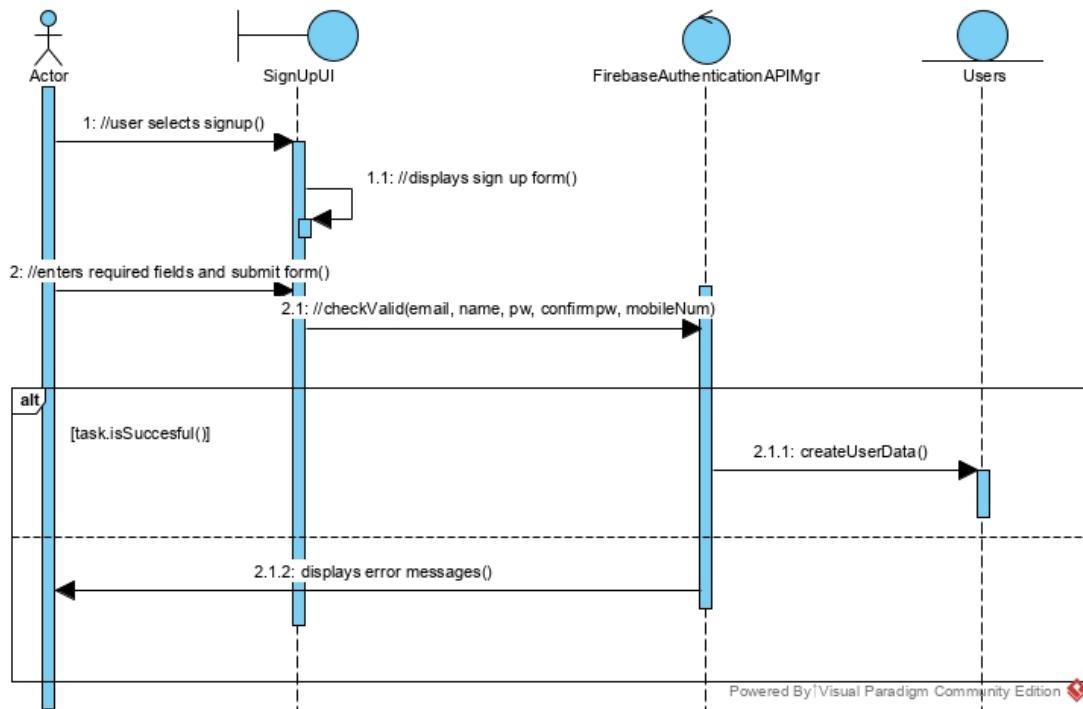
Extends:	NIL
Assumptions:	NIL
Notes and Issues:	

c. Class Diagram(tbd)

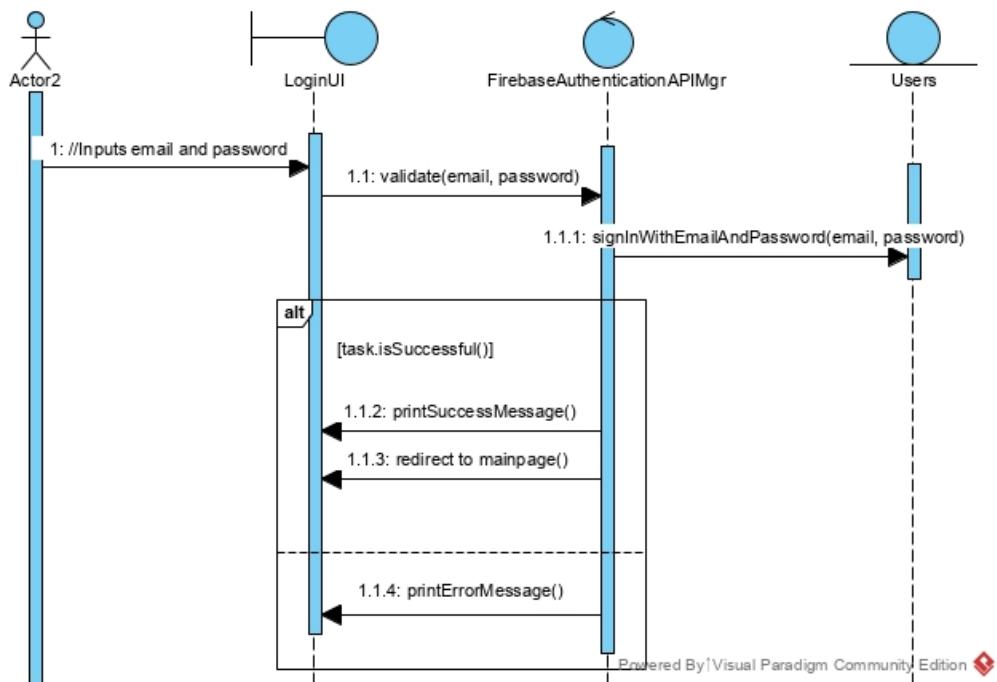


d. Sequence Diagram

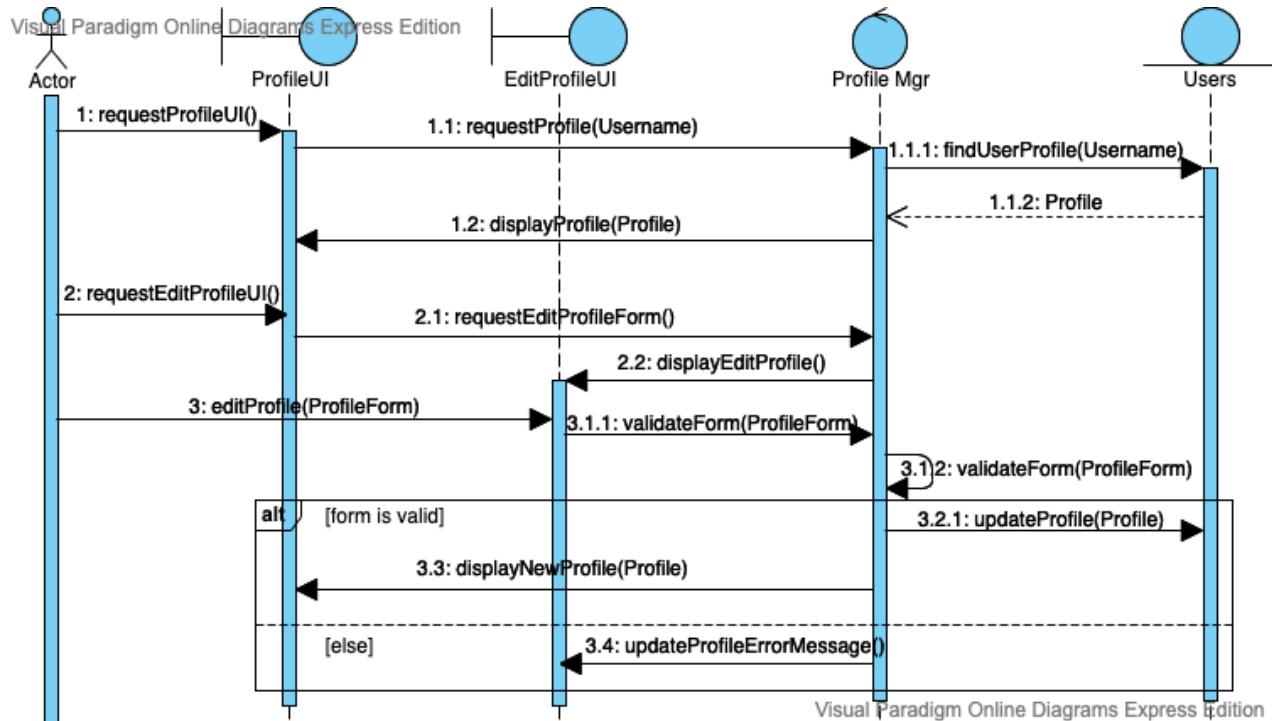
i. Account creation



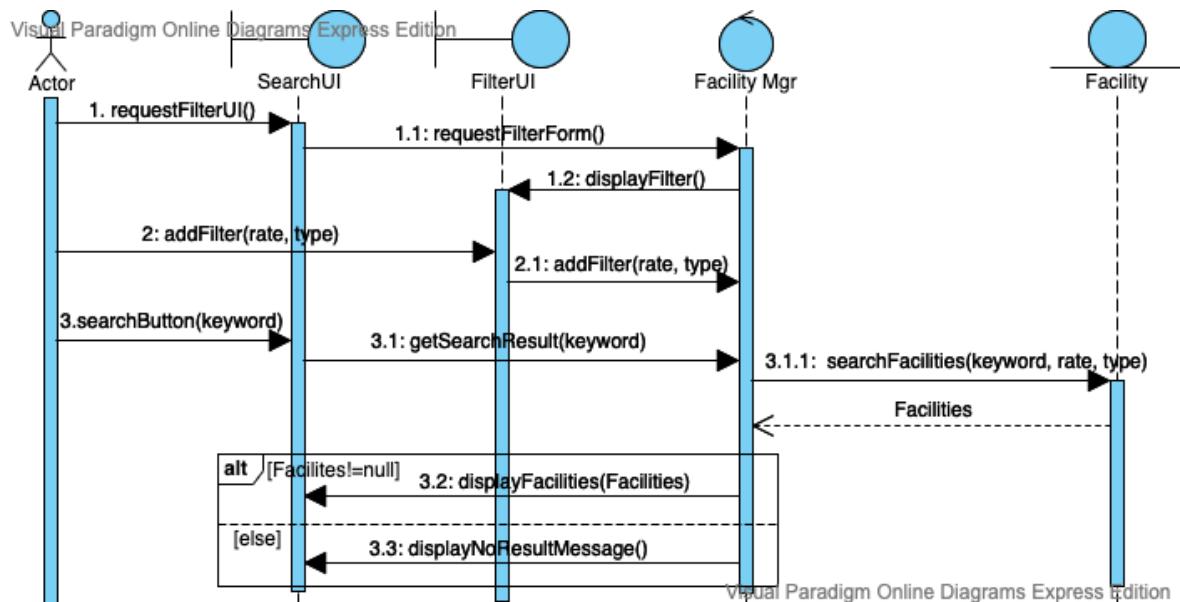
ii. Login



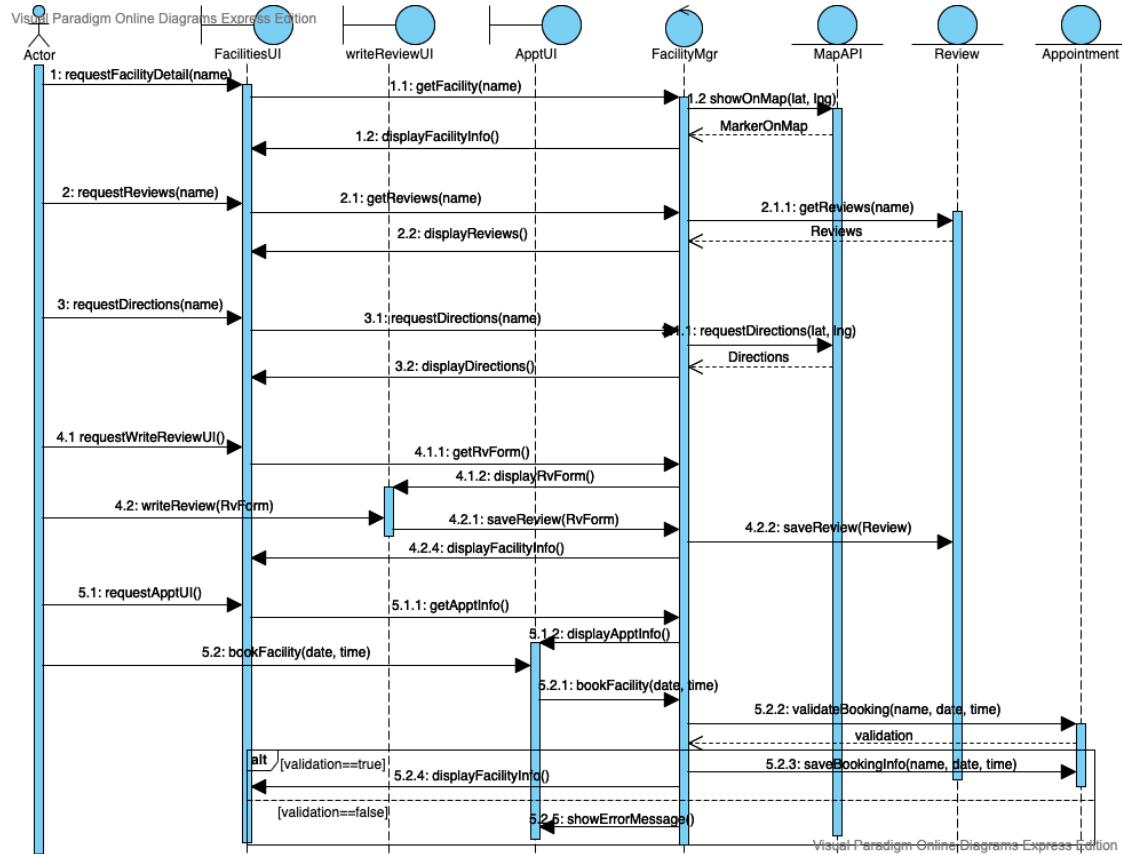
iii. View and Edit Profile



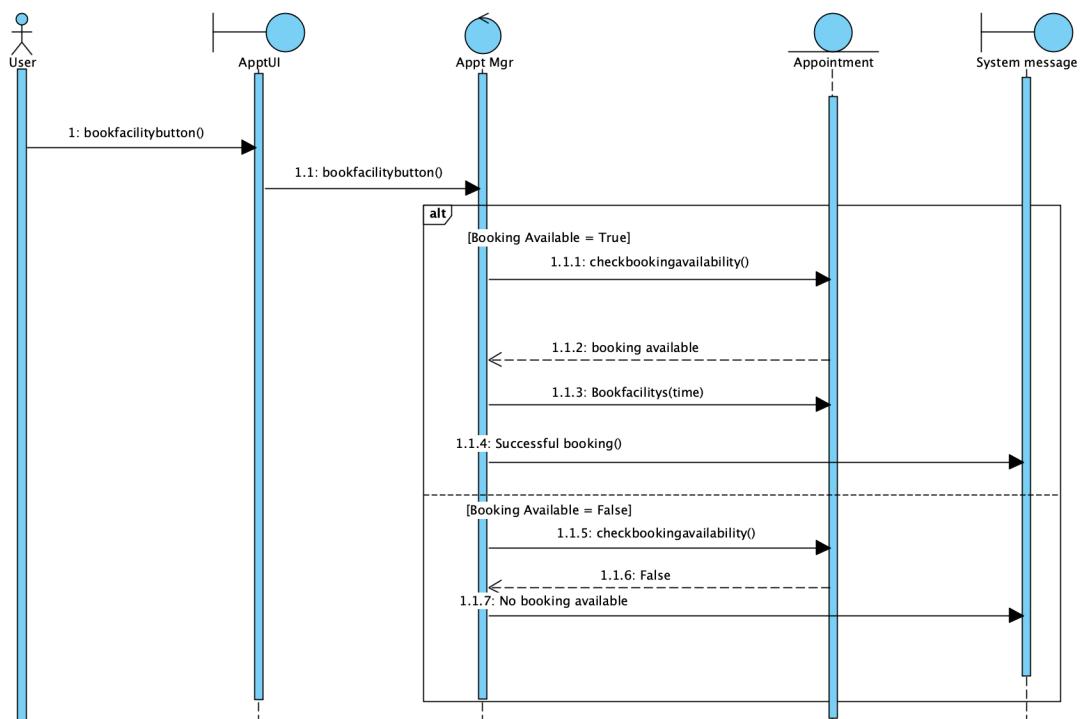
iv. Search



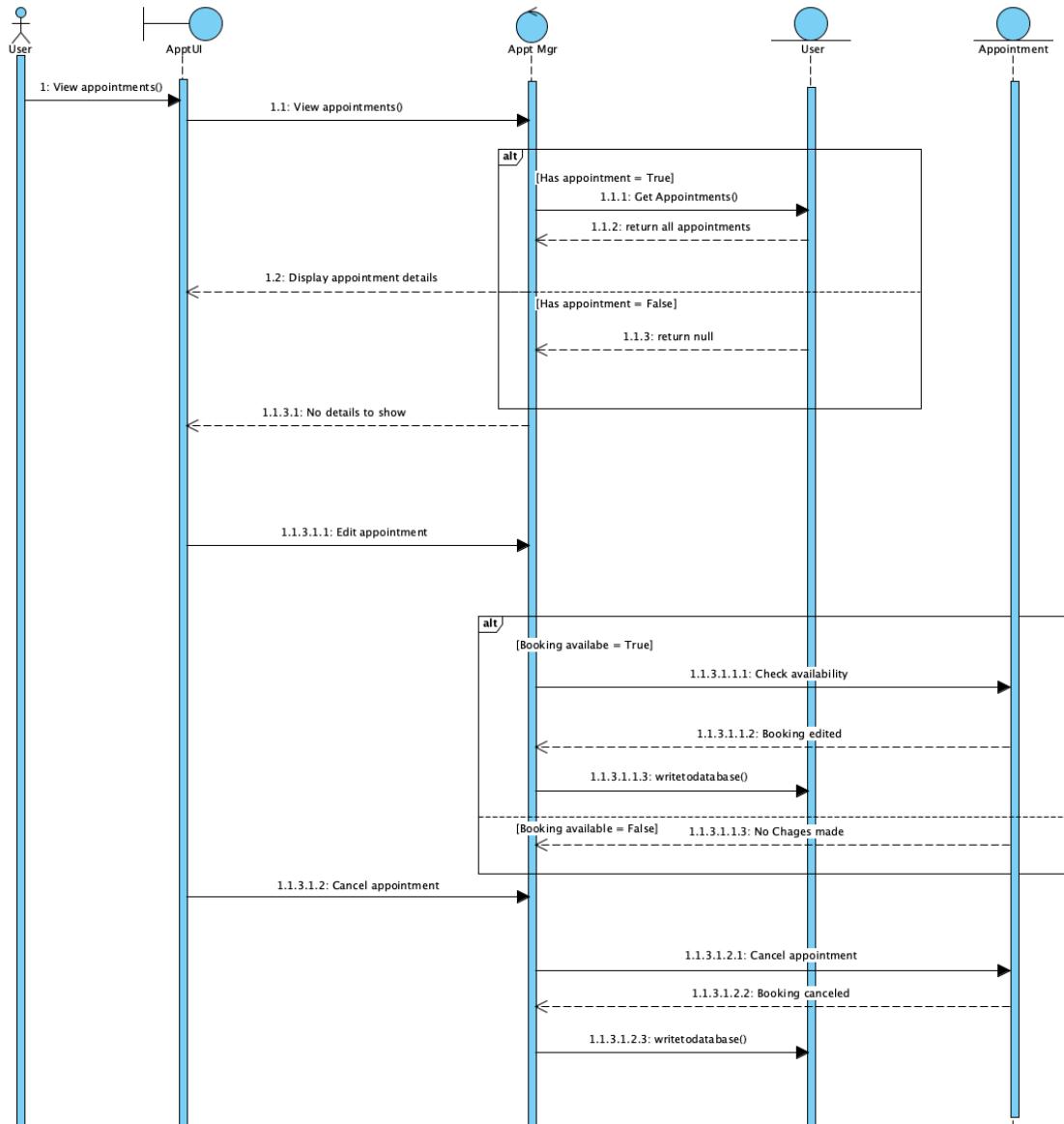
v. View facility



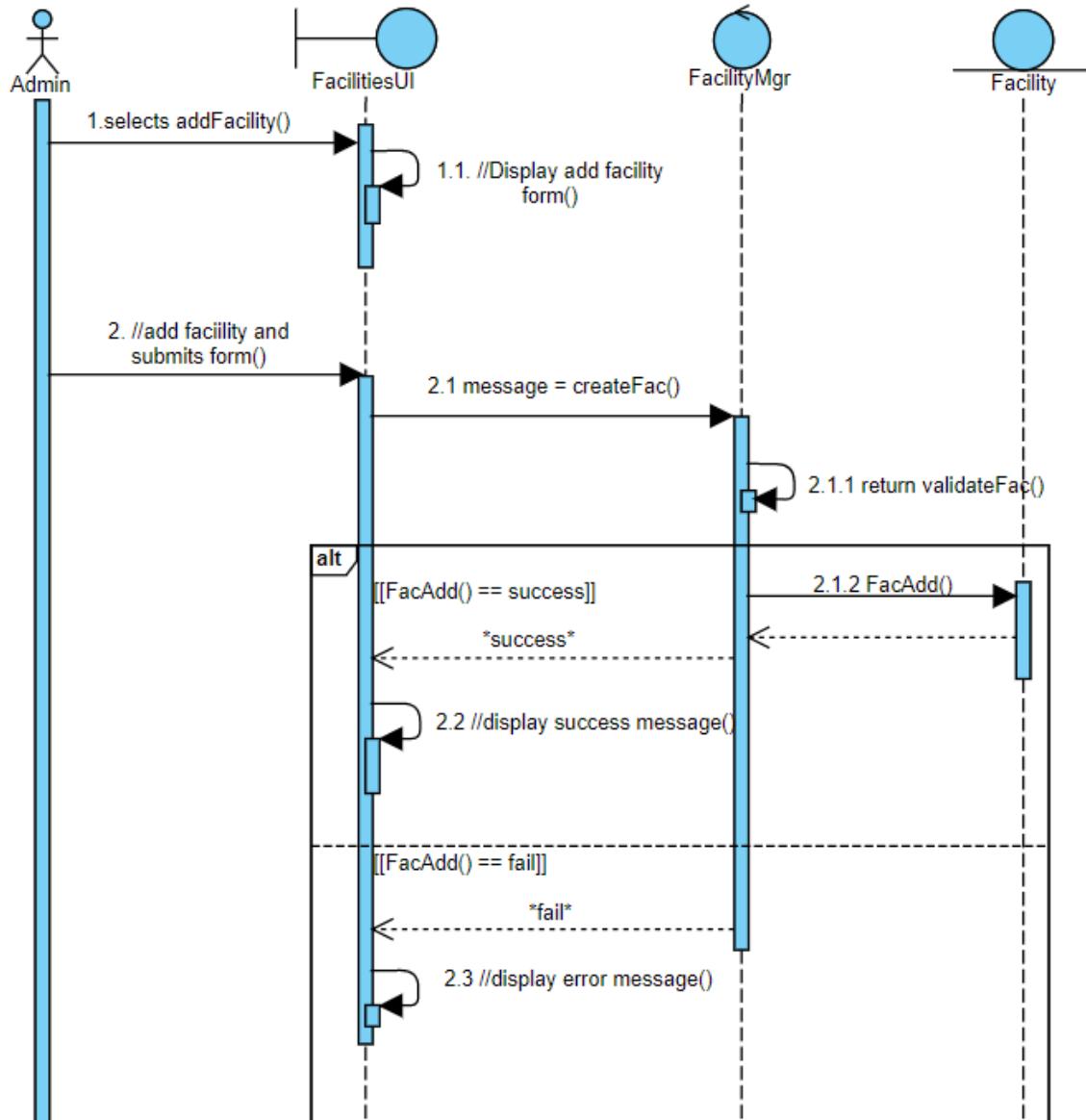
vi. Book appointment



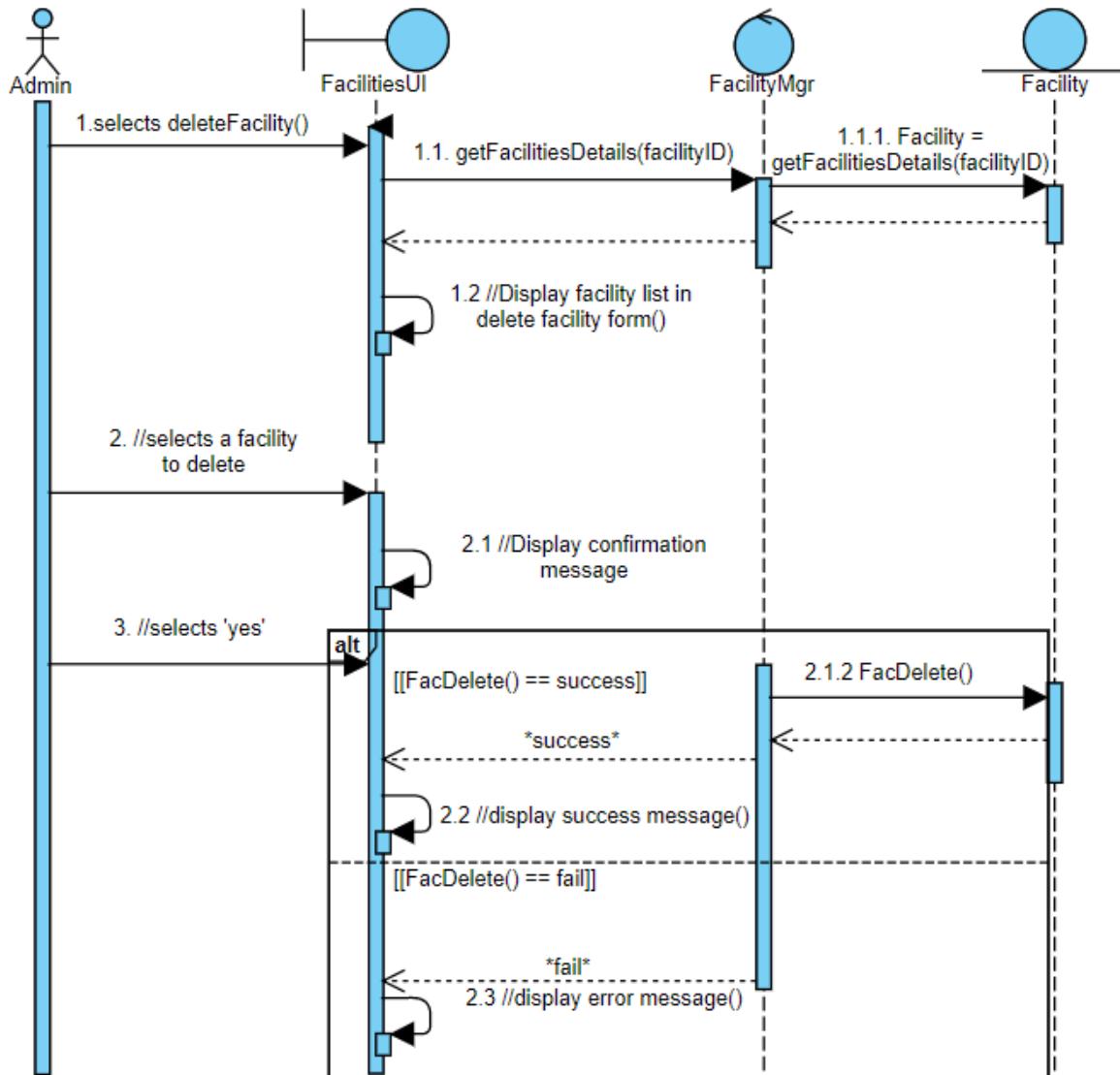
vii. View and edit appointment



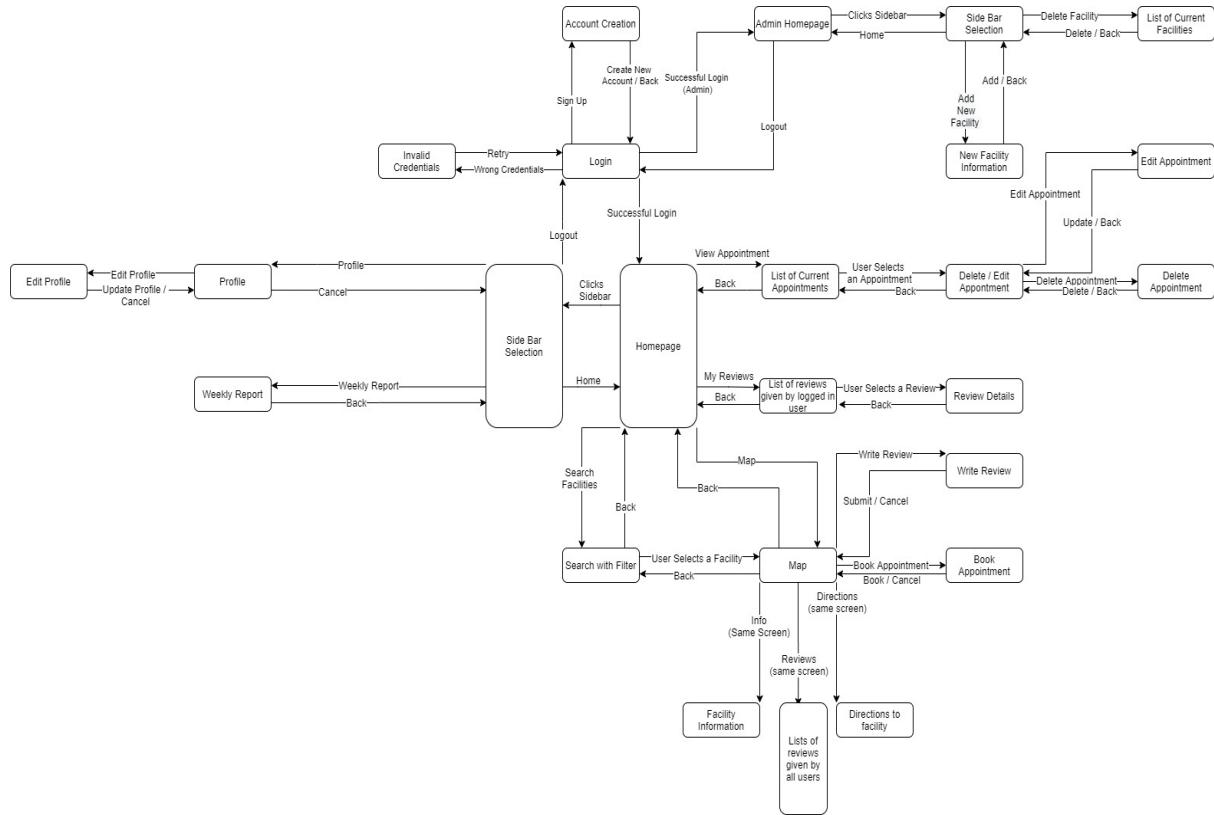
ix. Add facility



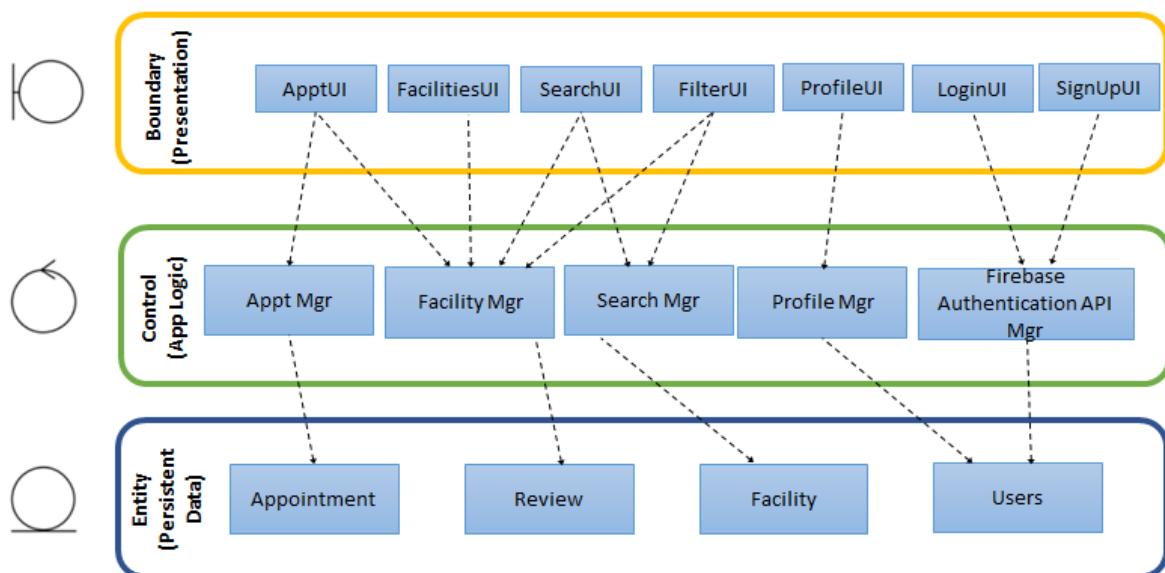
x. Delete facility



e. State Machine Diagram



f. System Architecture Diagram



g. Latest UI



Figure 1: Splash Screen

Figure 1 shows the screen the system displays while the app is launching. It is useful because the user is not left with a blank screen which can make the user think the app is not working properly and it will show the logo of the app to the user every time the app launches.

A composite image showing two side-by-side screens of a mobile application's sign-in and sign-up interface. The left screen is for 'SIGN IN' and the right screen is for 'SIGN UP'. Both screens share a common header with the 'SG SPORTS' logo and a tennis player icon. The 'SIGN IN' screen has input fields for 'email' and 'password', a green 'SIGN IN' button, and links for 'SIGN UP' and 'SIGN IN WITH GOOGLE'. The 'SIGN UP' screen has input fields for 'email', 'username', 'password', 'confirm password', 'gender' (with radio buttons for 'male' and 'female'), 'age' (with a dropdown menu showing '1'), and 'mobile number', followed by a green 'SIGN UP' button.

Figure 2-1: Sign in & Sign up

Figure 2-1 shows the user interface for sign in and sign up. The user can sign up for a new account by clicking the ‘SIGN UP’ button or sign into an already existing account by typing in email and password and clicking the “SIGN IN” button. The “SIGN IN” button is highlighted in bright color because this is the button that the user will use most of the time.

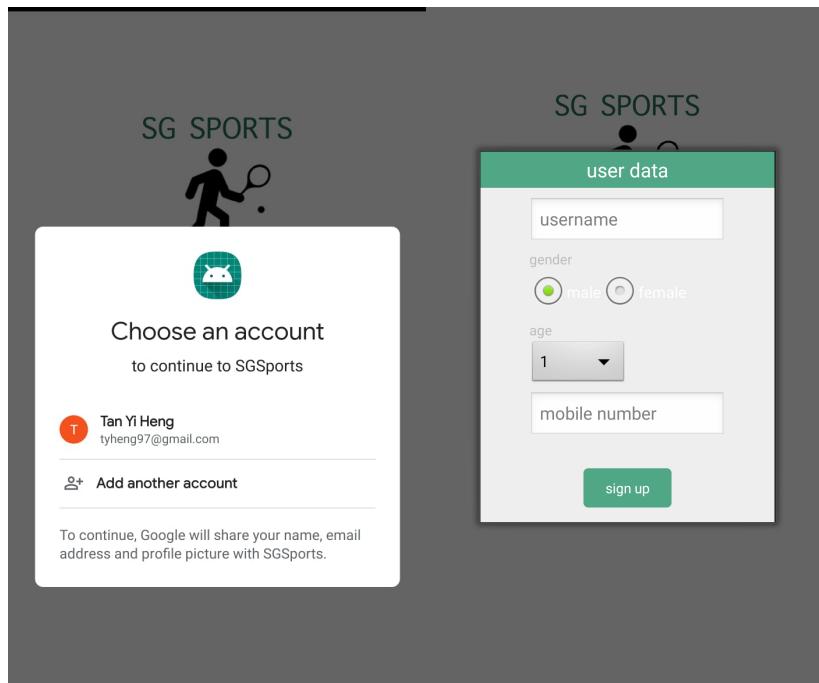


Figure 2-2: Sign in with google account

Figure 2-2 shows the screen that shows up when the user signs in with google account. If the user signs in with google account for the first time, the system displays a pop-up screen which requires the user to input more personal data.

The overall design for the authentication process is kept simple, and the layout is designed to be consistent with many other applications on the market to give users a sense of familiarity and prevent confusion.

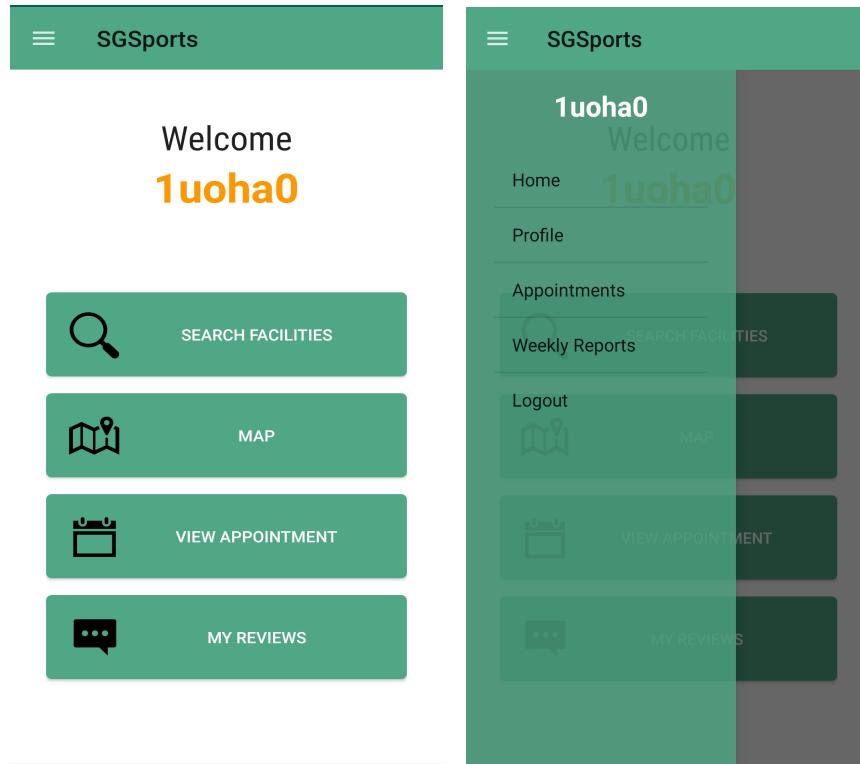


Figure 3: Main page & Sidebar

Figure 3 shows the main page of the application where the user can access the main functionalities of the application by four buttons: “SEARCH FACILITIES”, “MAP”, “VIEW APPOINTMENT”, and “MY REVIEWS”, each directing the user to a relevant activity. These buttons can be a guide to the user that tells what functionalities the app has when the user has no previous experience with it, and make it easier to access those functionalities.

The user can also open the sidebar on the left by clicking the hamburger button which sits on the left side of the app bar, as in any other applications. On the sidebar, by clicking one of the rows, the user can also access other functionalities regarding the user’s personal data. The sidebar disappears when the hamburger button is clicked again. The user can also use swipe motion to open (from left to right) and close (from right to left) the sidebar.

To achieve consistency, all components in this application such as the app bar and buttons use the same color scheme and fonts. Moreover, the app bar remains in the same position throughout the whole app cycle, so that the user can easily access other functionalities at any time.

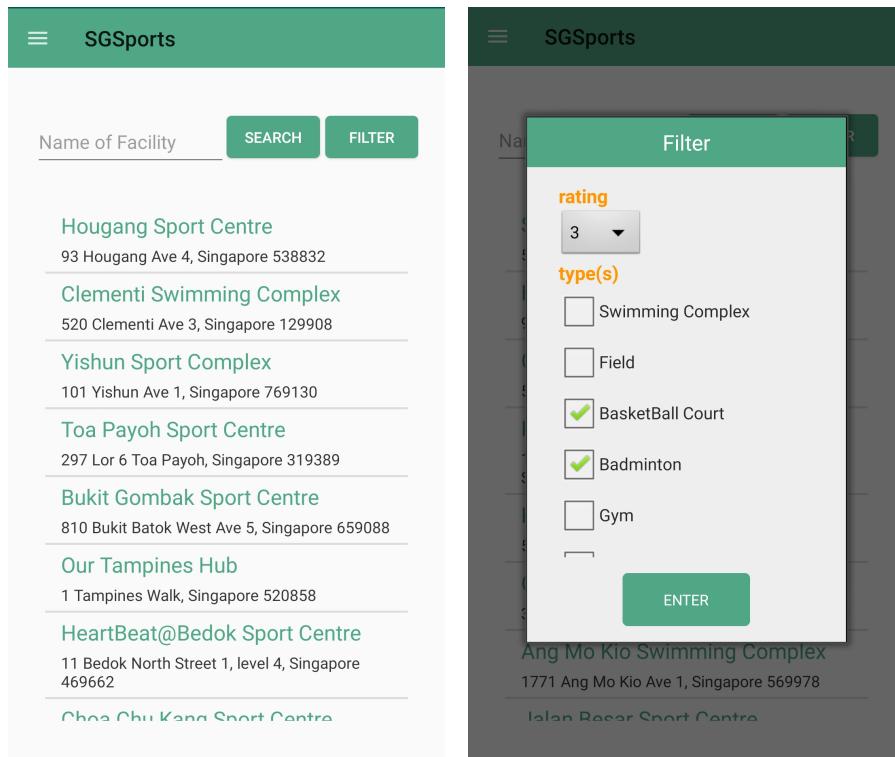


Figure 4: Facilities Search

Figure 4 shows the user interface for searching facilities that satisfy the user requirements. The system first displays the total list of facilities from the database. Each row contains the name and the address of the facility. Facility name is colored in light green so that the user can more easily identify the names and distinguish between rows.

When the user clicks the “FILTER” button, the system will display a pop-up screen where the user can set search conditions. After the user clicks “ENTER” to set the conditions and then clicks “SEARCH”, the system searches facilities that satisfy these conditions and whose names contain the keyword (if any), and prints out the result.

When the user clicks any row, the system will start the map activity where the user can view information, reviews, and location of that facility.

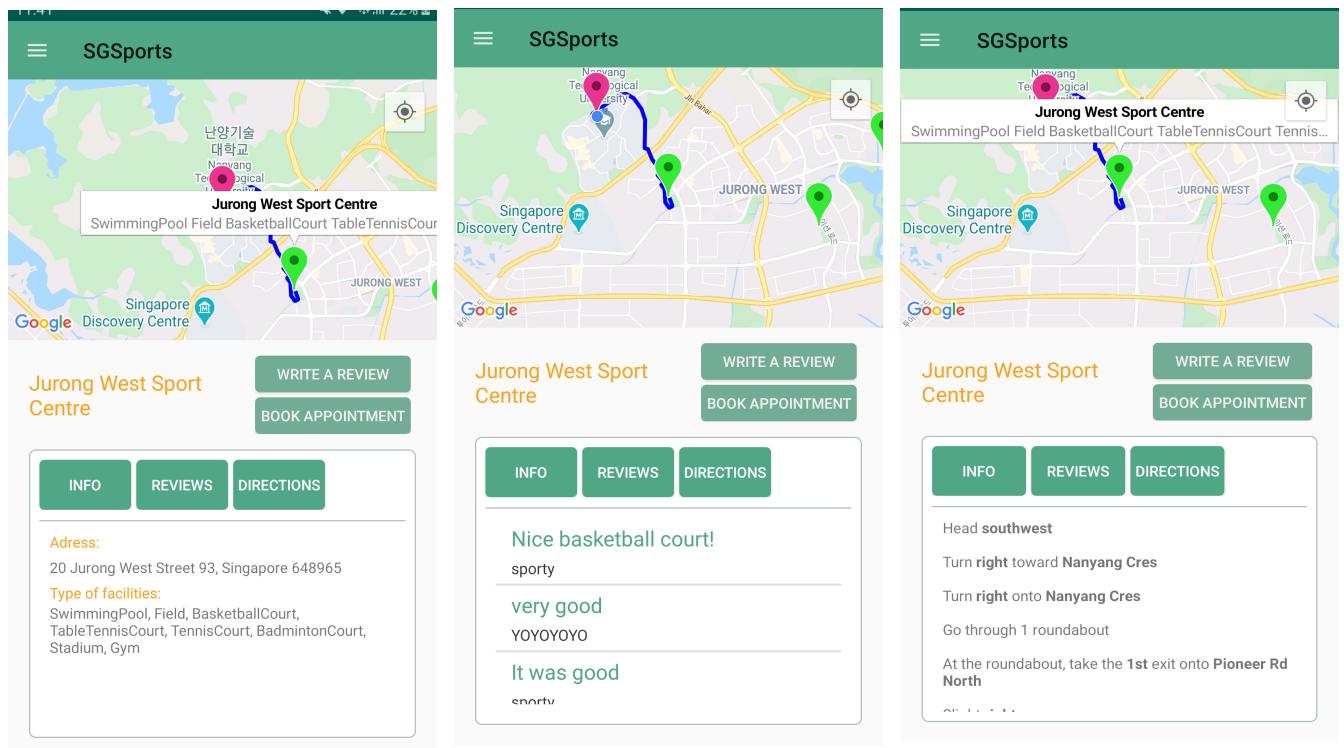


Figure 5: Map & View Facility

Figure 5 shows the page where the user can view information of the facility, write a new review of the facility, and book the facility.

If the user is directed to this page by clicking the “MAP” button on the home page, the system shows the user location on google map with a pink marker. Other green markers on map indicate facilities. By clicking a green marker, the map camera moves to the location of the facility and the map will also show the route to it from the user location. It will do the same when the user is directed to this page by clicking a facility on the search facility page.

The system also displays the facility name in yellow and its information in the layout below the name. In this layout, the user can switch from one section to another by clicking the corresponding button. Therefore, the user can view different information of the facility with the map above, not leaving the current page. On the “REVIEWS” section, a list of reviews of that facility is shown and by clicking each row, the user is directed to a different page where the details of the review can be found.

“WRITE A REVIEW” and “BOOK APPOINTMENT” buttons are separated from this layout and positioned to the right of the facility name. By clicking each, the system will direct the user to the corresponding page.

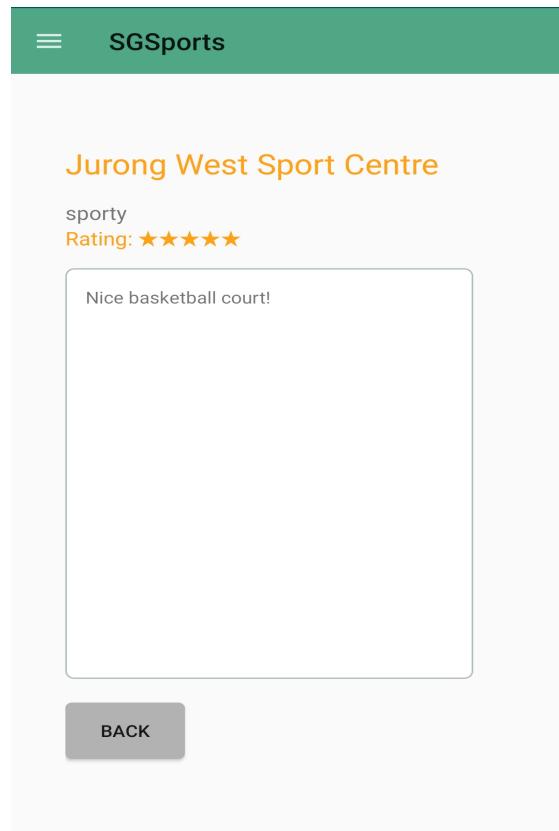


Figure 6: View Review page

Figure 6 shows the page where the user can read details of the review: the whole review text, writer, and the rating. By clicking the “BACK” button, the user can go back to the facility information page. The review text is written in a separate box for legibility.

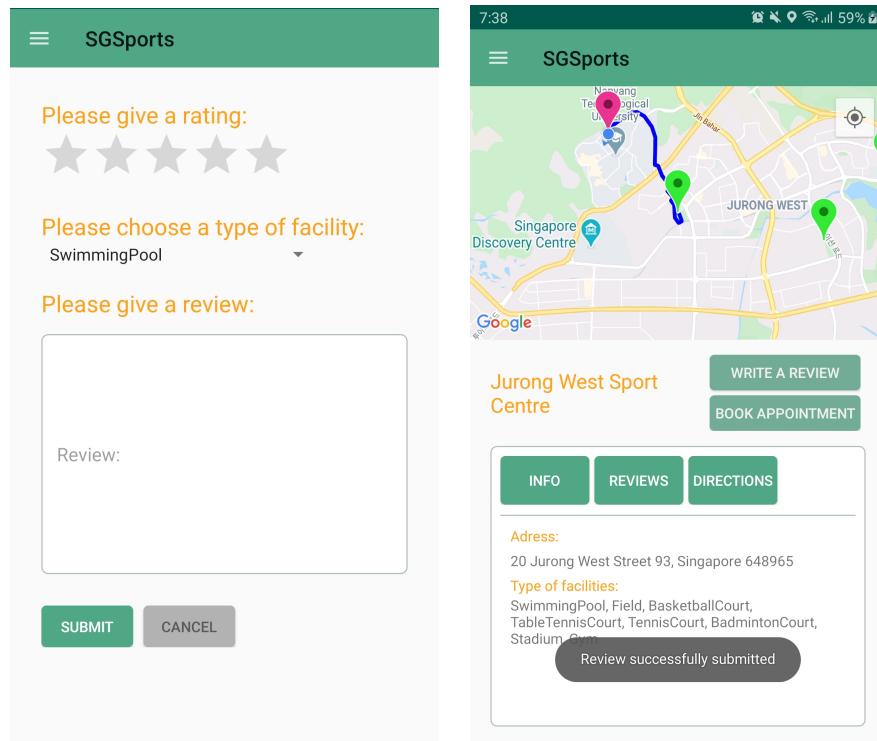


Figure 7: Write Review

Figure 7 shows a review form the user has to fill out. Rating is done by clicking the n-th(from the left) star (which is a commonly-used shape for rating) for a score of n. Then using a spinner, the user is required to choose a type of facility the user visited/used. Lastly, the user can write down more details in a separate text box.

The “SUBMIT” button is highlighted in a green color since submitting a review is the main function of this page. After submitting a review, the system will go back to the View Facility page and show a message to the user that the submission has completed. This page has the same format as the View Review page for consistency.

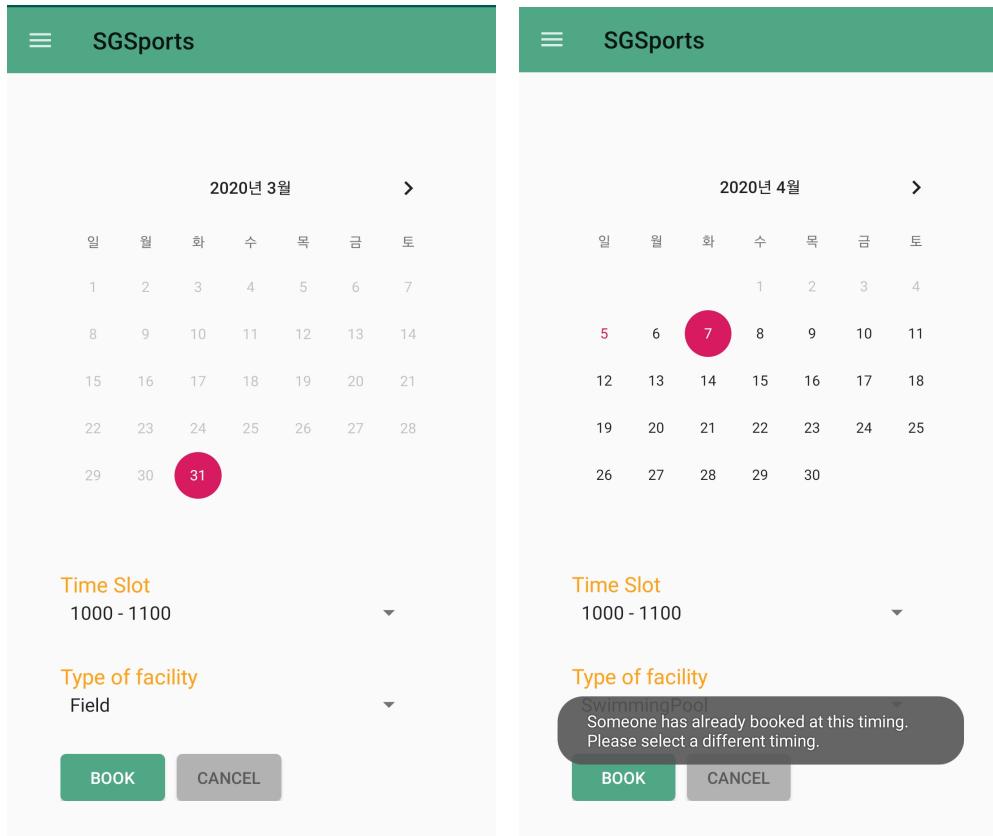


Figure 8: Book Facility

Figure 8 shows the user interface for booking a facility. The selected date is highlighted in a pink circle and the user can select one time slot and one type of facility the user wants to book using spinners.

The system verifies the booking when the user clicks the “BOOK” button. If the facility is already booked for that time and date, the system will fail to save the booking information, show an error message to the user, and stay on the same page. When the booking is successful, the system will also show another message to the user and go back to the View Facility page.

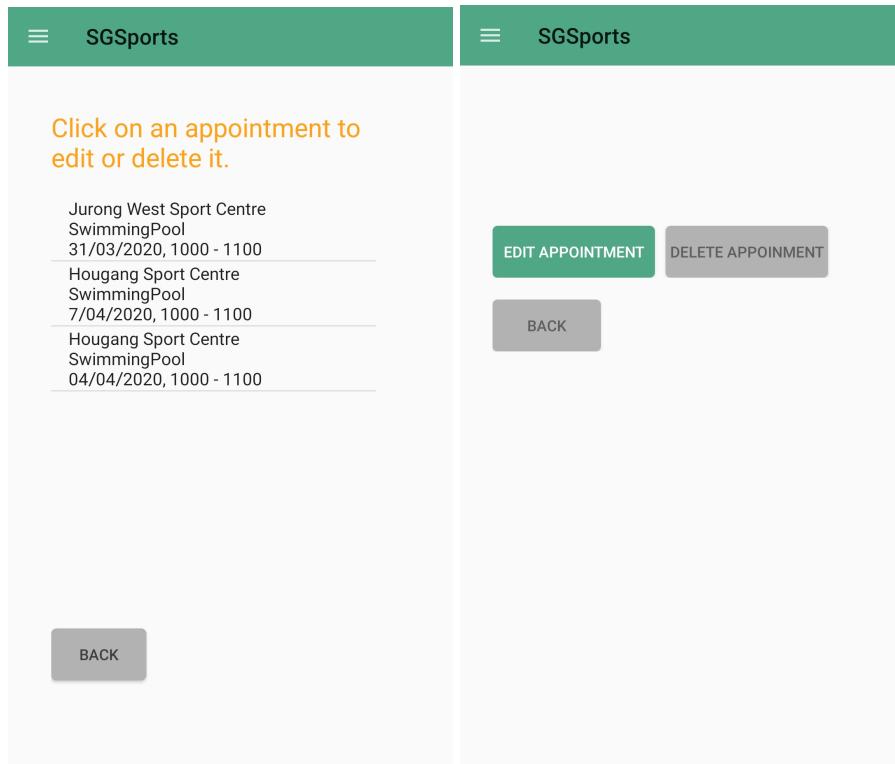


Figure 9: Manage Appointments

Figure 9 shows the user interface for managing the appointments. First, the user is given a list of appointments. By clicking each row, the user will be directed to the page (the right one) where the user can decide whether to edit or delete it by clicking each button.

The user can always go back to the former page by clicking the “BACK” button. When the user clicks the “EDIT APPOINTMENT” button, the user is directed to the page where the layout is the same as the Book Facility page but with different activity.

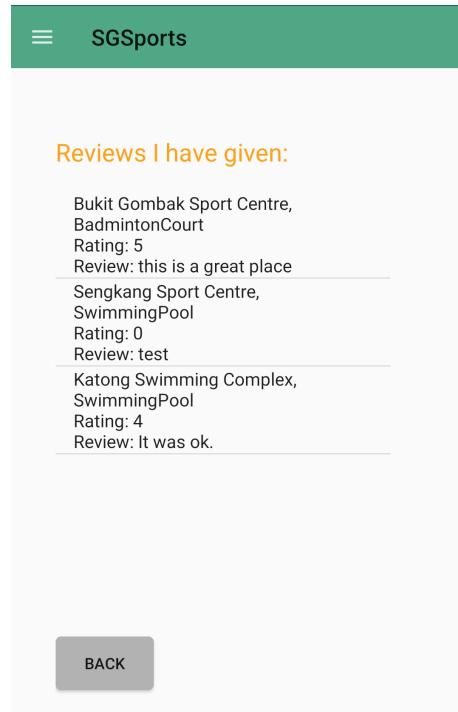


Figure 10: View My Reviews

Figure 10 shows the page where the user can view the whole list of reviews that the user has written. By clicking each row, the user is directed to the View Review page for that review. This page has the same layout as Manage Appointment page for consistency.

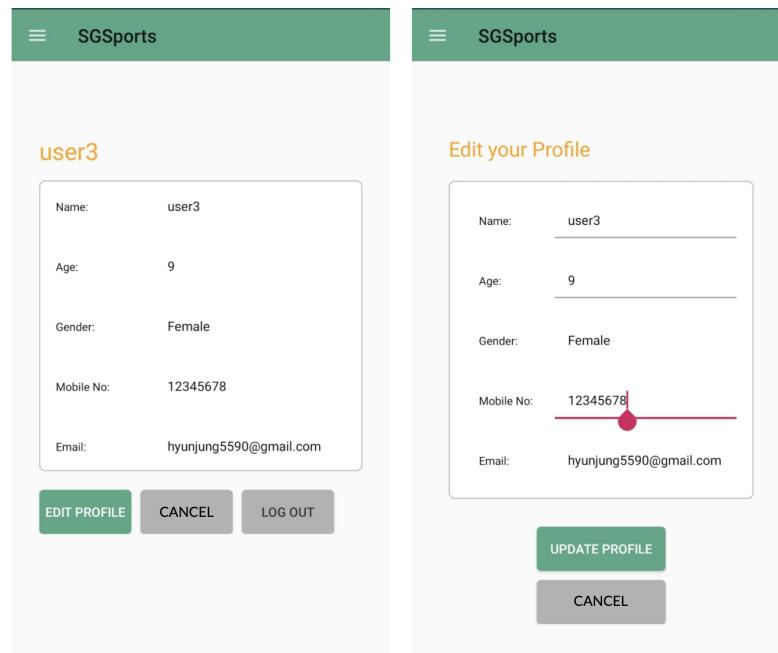


Figure 11: View/Edit Profile

Figure 11 shows the user interface for managing the user profile. As on the left page, the user can view the current profile and choose to edit the profile, go back to the previous page, or log out from this account. Since the main function of this page is editing the profile, the “EDIT PROFILE” button is highlighted in green.

As on the right page, the user can modify user name, age, and mobile number fields. For consistency, the layout remains the same as the View Profile page. If the user clicks the “UPDATE PROFILE” button and the newly filled out form doesn’t satisfy the requirements, the system will display an error message and remain on the same page. If it does, the system will save the new profile in the database and go back to the View Profile page so that the user can immediately check out the changes that are made.

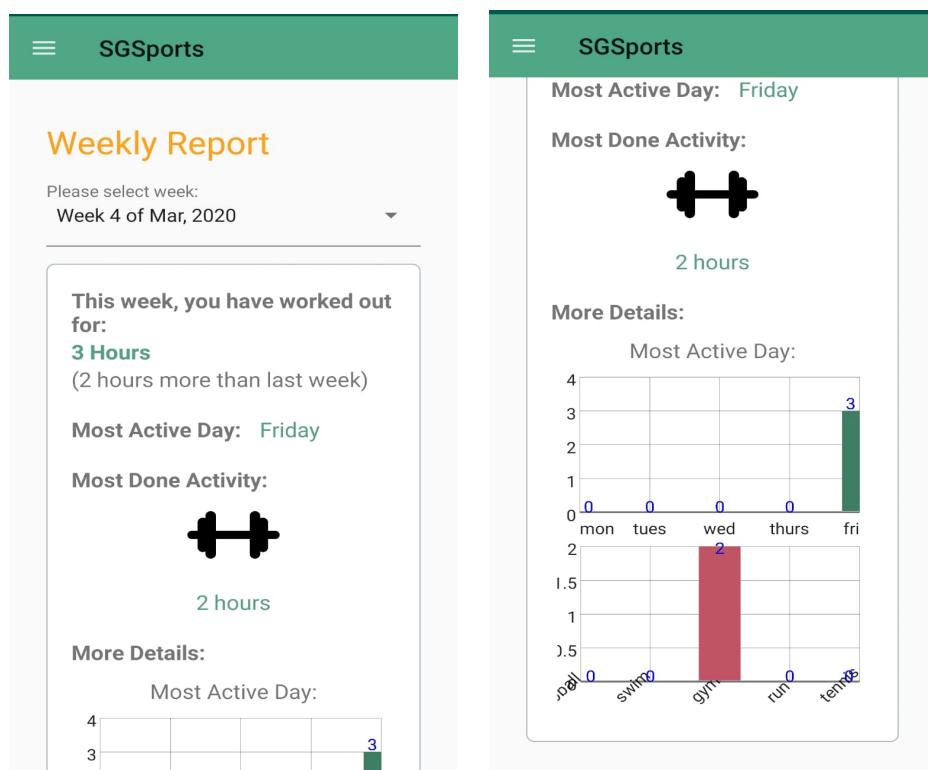


Figure 12: Weekly Reports

Figure 12 shows the user interface for the system-created weekly reports of the user’s sports activities. The user can select any past week from the spinner that is below the title “Weekly Report” and the corresponding reports will be shown below. The reports include the total workout time for the week, the most/least active day with the total time, and the most & least done activities also with the total time spent.

For legibility, the texts that change for each week(ex: total time) are highlighted in green color and icons are used. And the user is also provided with graphs that show how many hours the user had spent on each sports activity and also on each day.

Since the length of the reports layout vary with weeks, the page is scrollable.

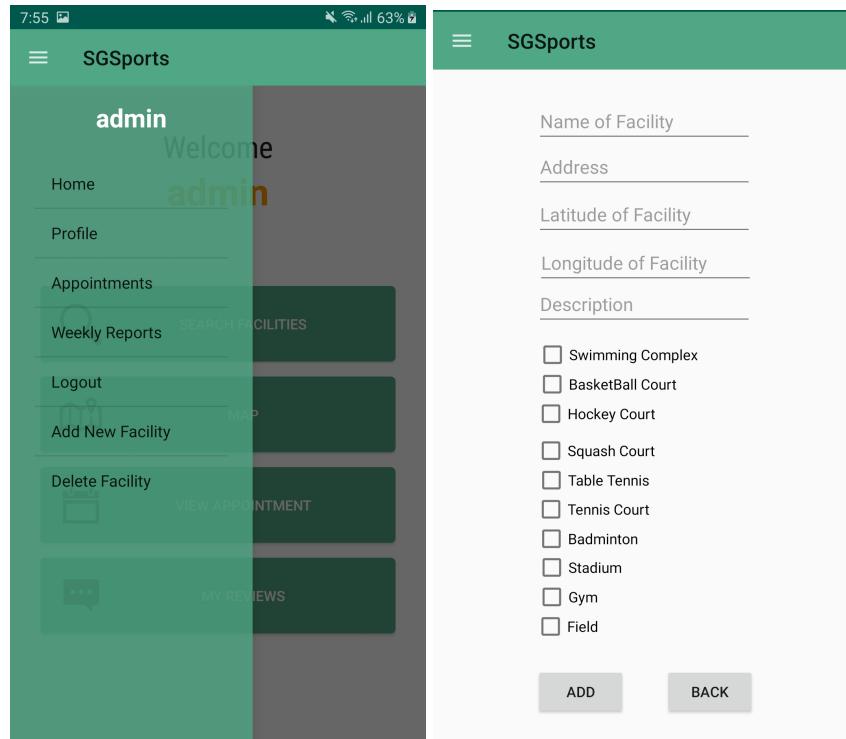


Figure 13: Add Facility

Figure 13 shows the user interface for adding a new facility. This page is only accessible by the “admin” user, and can be accessed from the sidebar. The sidebar shows two more rows of “Add New Facility” and “Delete Facility” only when the current user is the “admin”.

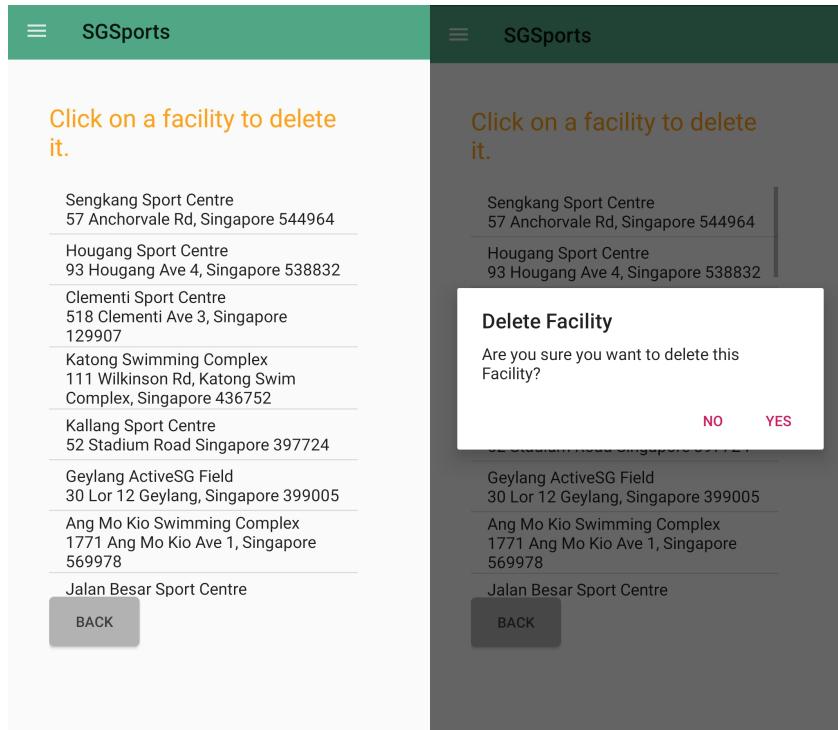


Figure 14: Delete Facility

Figure 14 shows the user interface for deleting an existing facility. As with the Add Facility page, this page is only accessible by the “admin” user and through the sidebar. The “admin” user can scroll through a list of all facilities and click one of them to delete. Before the system deletes the facility from the database, it confirms with the user again by showing a dialog as above.

3. NON-FUNCTIONAL REQUIREMENTS

Flexibility:

1. Provisions shall be made for the future usage of multiple languages.

Maintainability:

1. The system shall not be under maintenance more than once in a 24-hour period.

Performance:

1. The application must support at least 100 unique user accounts
2. The response time for menu navigation must be under 1 second
3. The response time for generating the search results must be under 5 seconds.
4. The time taken for the system to publish a new review must be under 10 seconds.
5. The time taken to update the rating of the sport facilities which has a new review published must be under 10 seconds.

Reusability:

1. The application must be able to accommodate the searching of another type of facility (eg. dental clinics).

Reliability:

1. The application must be able to be fully displayed within 3 seconds upon starting up.
2. The user must be able to login successfully upon submission of the correct username and password.
3. The application will be regularly updated to ensure that the list of sporting facilities is accurate.

4. DATA DICTIONARY

Term	Definition
System	The System refers to the application that we have created. This application provides features such as finding the nearest sporting facility, managing user profile etc.
User	Human User of the system who is registered. They have access to features like managing, user profile.
Admin	Human User of the system who is registered. They have additional access and privileges as compared to Users, so that they can manage the whole system.
User Interface	Allows the User to interact with the system in an intuitive way.
User Profile	A summary of a Users data, which can be printed. Modifiable data should also
Map	A graphical representation of an area showing features such as roads, sport facilities and shortest paths.
Global Positioning System (GPS)	A function utilised in the application to retrieve a user's exact location.
Report	A printable document of data, for example a summary of performed activities during a week.
Advanced Encryption Standard (AES)	A reliable and well established encryption algorithm that supports key sizes of at least 128 bit.