
Software Requirements Specification

for

Skyscape

Version 1.0 approved

Prepared by

LI YAOHAO,
LIM ZHI YONG,
JIN YING,
KIERAN LIOW,
KIERSTEN YEO SHU XIAN,
YOON SANG WON

NANYANG TECHNOLOGICAL UNIVERSITY TEAMWON

12 April 2024

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	2
1.5 References	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	3
2.3 User Classes and Characteristics	3
2.3.1 General Users	3
2.3.2 Photographers:	4
2.4 Operating Environment	4
2.5 Design and Implementation Constraints	4
2.6 User Documentation	4
2.7 Assumptions and Dependencies	5
3. External Interface Requirements	6
3.1 User Interfaces	6
3.1.1 Login	6
3.1.2 Register	6
3.1.3 Home Page	7
3.1.4 Sort Home Page	7
3.1.5 View Details	8
3.1.6 Search Users	8
3.1.7 Search Location	9
3.1.8 For You Feed	10
3.1.9 Following Feed	10
3.1.10 Upload Photo	11
3.1.11 Profile	11
3.1.12 Account Details	12
3.1.13 Edit Profile	12
3.1.14 Personal Feed	13
3.1.15 Report Bug	13

3.2 Hardware Interfaces	14
3.3 Software Interfaces	14
3.4 Communications Interfaces	14
4. System Features	15
4.1. Use case of Registration	15
4.1.1 Functional Requirements	16
4.2. Use case of Login	16
4.2.1 Functional Requirements	17
4.3. Use case of Access Homepage	18
4.3.1 Functional Requirements	18
4.4. Use case of Edit Homepage	19
4.4.1 Functional Requirements	20
4.5. Use case of Calendar Function	20
4.5.1 Functional Requirements	21
4.6. Use case of Location Details	22
4.6.1 Functional Requirements	22
4.7. Use case of Sort Locations	23
4.7.1 Functional Requirements	23
4.8. Use case of Upload Pictures	24
4.8.1 Functional Requirements	25
4.9. Use case of Follow User	25
4.9.1 Functional Requirements	26
4.10. Use case of View and Edit Following	27
4.10.1 Functional Requirements	27
4.11. Use case of Account Settings	28
4.11.1 Functional Requirements	29
4.12. Use case of Report Bug	30
4.12.1 Functional Requirements	30
4.13. Use case of View All Photos Uploaded	31
4.13.1 Functional Requirements	32
4.14. Use case of View Photos by Following	32
4.14.1 Functional Requirements	33
4.15. Use case of Access other User's Profile	34
4.15.1 Functional Requirements	34
4.16. Use case of Access Personal Profile	35
4.16.1 Functional Requirements	35

5. Other Nonfunctional Requirements	36
6. Other Requirements	37
6.1. Black Box Testing (highlight indicates the invalid input out of multiple inputs)	37
6.1.1. Login Functionalities	37
6.1.2. Register/Sign up Functionalities	38
6.1.3. Add location Functionalities	40
6.1.4. Upload Feed Functionalities	41
6.1.5. Follow Users Functionalities	42
6.1.6. Edit Profile Functionalities	43
6.1.7. Report Bugs Functionalities	45
6.2. White Box Testing	46
6.2.1. Register New Account	46
6.2.2. Upload Pictures	47
7. Appendix A: Glossary	48
7.1. Data Dictionary	48
7.2. Sunset Quality Algorithm	50
8. Appendix B: Analysis Models	51
Complete Use Case Model	51
Complete Class Diagram	52
Complete Dialog Map	53
Complete Application Skeleton	54
Key Boundary and Control Classes Diagram	55
Complete Sequence Diagrams	56
Register	56
Login	57
Upload Pictures	58
Follow User	59
AccessHomePage	60
ViewDetails	61
AddFavouriteLocation	62
RemoveFavouriteLocation	63
BugReport	64

Revision History

Name	Date	Reason For Changes	Version
LI YAOHAO	10/04/2024	First Draft	1.0.0

1. Introduction

1.1 Purpose

This document specifies the software requirements for Skyscape V1.0, a mobile application. It includes the requirement specifications, functional requirements, non-functional requirements, user interfaces and other related documentation.

Skyscape is a mobile application that is developed using Flutter and Firebase to achieve its intended use of providing accurate information about sunset and allowing users of the application to form an online community through the app.

1.2 Document Conventions

This document uses the following conventions.

Software Requirement Specification Format: IEEE standard. Priorities of higher level requirements are inherited by detailed level requirements.

Heading 1: Font Family: Times, Font Size: 18, Font Weight: Bold

Heading 2: Font Family: Times, Font Size: 14, Font Weight: Bold

Heading 3: Font Family: Times, Font Size: 12, Font Weight: Bold

Content: Font Family: Arial, Font Size: 11

1.3 Intended Audience and Reading Suggestions

This document is intended for all the users, developers, project managers, testers and other supporting roles that will require information about this mobile application. It includes the purpose of the app and how it can be used for everyday users, as well as the requirements and functionalities needed for the developers and managers and testers.

Users: Users should refer to Section 2.2 for Product Functions as well as Section 3.1 to familiarize themselves with the user interfaces.

Developers and Product Managers: Highly recommend to go through the entire document from start to finish to ensure they have a good understanding of the application.

Testers: Testers should focus on Section 2.2 for Product Functions and Section 4 for the System Features in order to devise suitable test plans.

1.4 Product Scope

Skyscape is a mobile application that is developed using Flutter and Firebase to achieve its intended use of providing accurate information about sunset and allowing users of the application to form an online community through the app.

In today's fast-paced world and especially in the hectic society that we are living in, finding moments of tranquility and peace is more important than ever for our mental health and well-being. The simple act of taking in the sight of a beautiful sunset has been scientifically proven to trigger positive emotions which can improve one's mood, making it ideal after a tiring day of work or studies. The anticipation of a beautiful sunset will encourage many to take a break from their busy schedules and go out into nature either by themselves or with their loved ones, boosting physical activity while at the same time serving as a valuable opportunity to unwind and distress by immersing in the present.

SkyScape is not only for the casual users who occasionally enjoy sunsets, but it also caters to photography enthusiasts and nature lovers by providing them an intuitive interface to upload and share photos of sunsets taken at any given day.

The objective set out for SkyScape is to enrich people's lives with a much needed moment of awe and relaxation that can be easily achieved in this fast-paced world.

With the development of this application, SkyScape aims to promote both physical and mental well-being by encouraging people to go out and chase the sunset and sunrise, relieving their stress in a fast-paced society.

1.5 References

- I. Source Code: <https://github.com/zhiyongqqqqq/Skyscape>
- II. Firebase: <https://firebase.google.com/>
- III. Flutter: <https://flutter.dev/>
- IV. Open-Meteo API : <https://open-meteo.com/>
- V. Pollutant Standards Index (data.gov.sg) :
[https://beta.data.gov.sg/collections/1396/datasets/d_8a7850dc3993dc45f1620b9972c58d4d
/view](https://beta.data.gov.sg/collections/1396/datasets/d_8a7850dc3993dc45f1620b9972c58d4d/view)

2. Overall Description

2.1 Product Perspective

The description within this SRS is for a completed mobile application that does not belong to another product family.

2.2 Product Functions

SkyScape aims to provide accurate information about current and upcoming status of sunsets and has the following features:

1. Users can register for an account if they do not have one and can login to their registered account.
2. Users can access the home page to view the sunset quality of the bookmarked locations.
3. Users can add and remove certain locations from the list of “favourited locations” for ease of viewing.
4. Users can view sunset quality details for up to 7 days ahead.
5. Users can view in depth information regarding the sunset for a chosen location.
6. Users can sort the home page either by alphabetical order of the location names, or by the sunset quality.
7. Users can upload pictures of the sunset for other users to view.
8. Users can follow other users on the application to be able to view the pictures they have uploaded.
9. Users can view and edit the list of users that he/she is currently following.
10. Users can edit his/her account details in the account settings.
11. Users can report a bug that is found on the application.
12. Users can view all the pictures uploaded by all users on the application, or choose to only view pictures posted by the accounts that he/she is following.
13. Users can view other user's accounts and see the pictures posted by them on their profile.

2.3 User Classes and Characteristics

2.3.1 General Users

Aspect	Description
Frequency of Use	High
Subset of Product Functions Used	All
Technical Expertise	1. Comfortable with using mobile applications.
Age	All
Characteristics	<ol style="list-style-type: none"> 1. Looking for external motivation to seek moments of tranquility, peace and achieve mental well-being. 2. Looking to take a break from busy schedules and go out into nature to unwind and distress by immersing in the present.

2.3.2 Photographers:

Aspect	Description
Frequency of Use	Medium
Subset of Product Functions Used	All
Technical Expertise	1. Comfortable with using mobile applications.
Age	All
Characteristics	1. Looking for information on weather related information that could impact the quality of photos taken. 2. Looking to share the photos taken with other users.

2.4 Operating Environment

Skyscape was developed to function in ARM64 CPU and Android Version 14.0 and above, currently, it does not support iOS.

The development environment for the application is as such:

Front-end: Flutter

Back-end: Firebase

2.5 Design and Implementation Constraints

1. The database used to manage all user related data is Firebase, thus, the working state of the application will ultimately also depend on the status of Firebase.
2. The meteorological information obtained within the application is done through an API call. Thus, the degree of functionalities and accuracy to which the application can achieve, is dependent on the status of API.
3. The pollutant standards index obtained within the application is done through an API call. Thus, the degree of functionalities and accuracy to which the application can achieve, is dependent on the status of API.
4. The maintainability of the software will be under the responsibilities of the development team that developed this application, unless it has been acquired by other companies.

2.6 User Documentation

Information on running Skyscape is provided in the README.md file in the GitHub repository. The source code for the latest version of Skyscape can be found on <https://github.com/zhiyonggggg/Skyscape>.

2.7 Assumptions and Dependencies

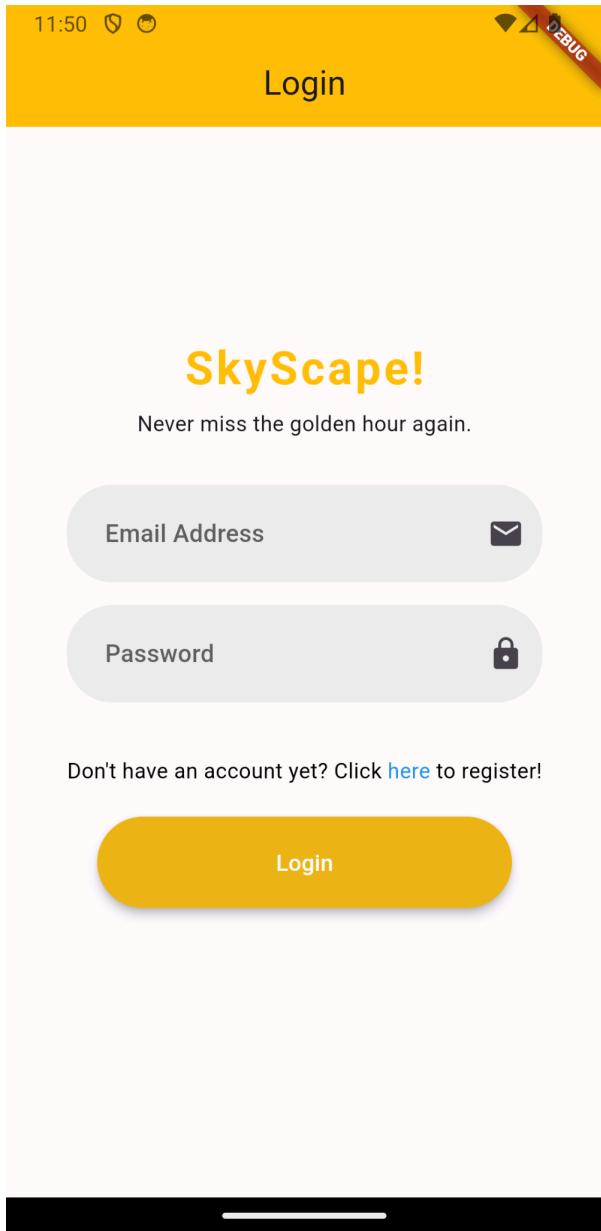
1. The meteorological information obtained within the application is done through two API calls, they are:
 - a. Data.gov.sg for information about Pollutant Standards Index
 - b. Open-Meteo for all other meteorological information

It is assumed that the information provided by these 2 APIs are accurate.

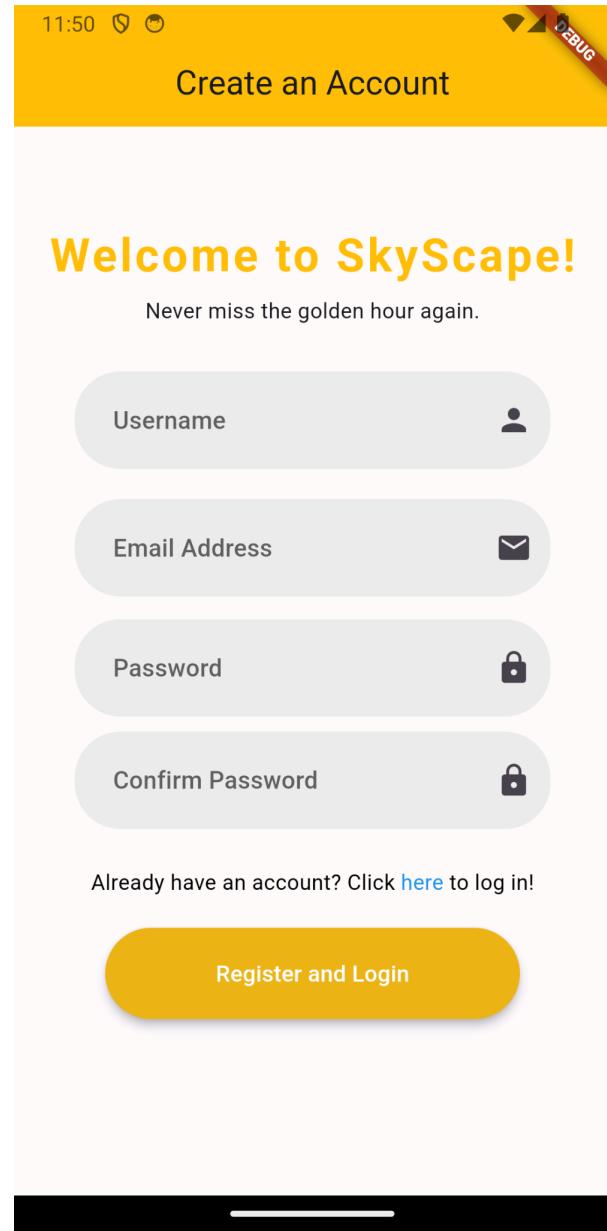
3. External Interface Requirements

3.1 User Interfaces

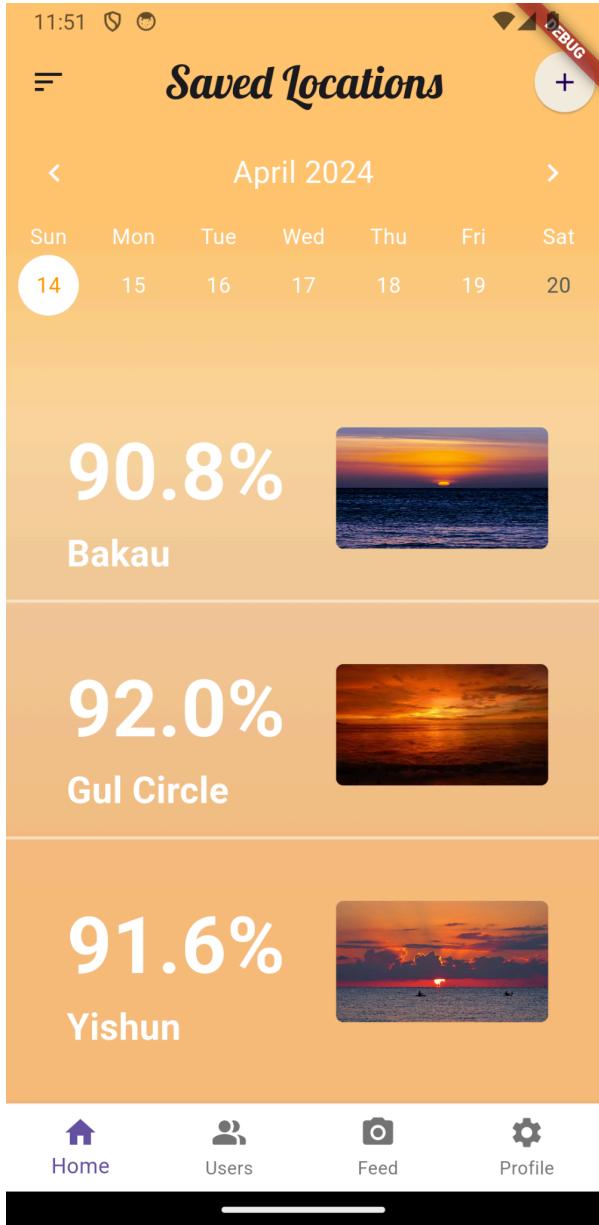
3.1.1 Login



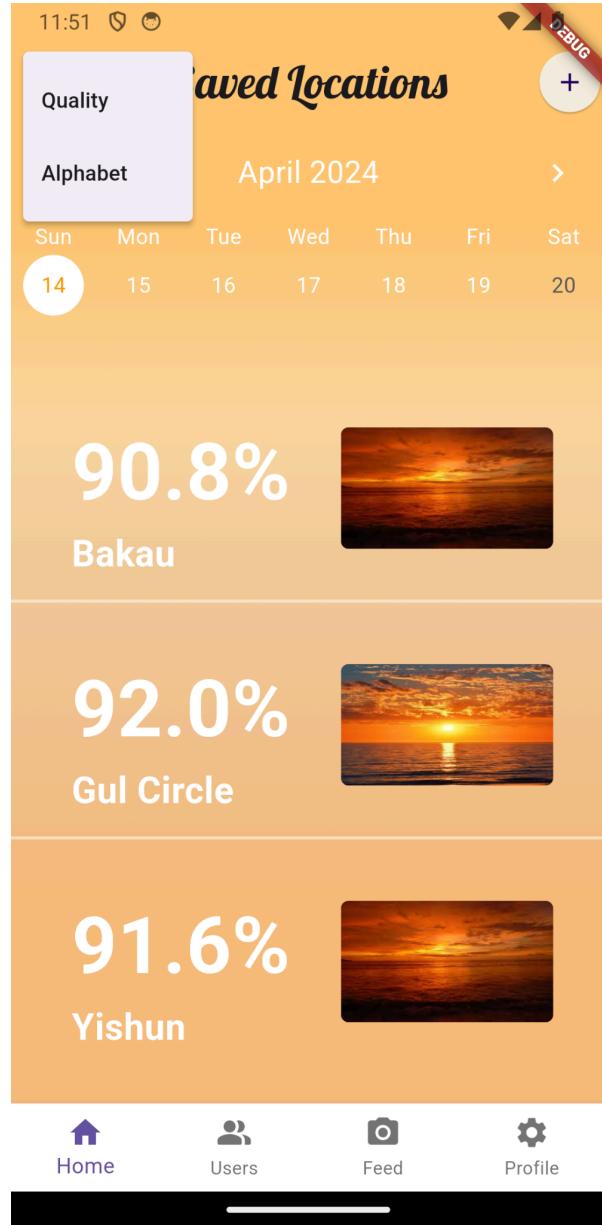
3.1.2 Register



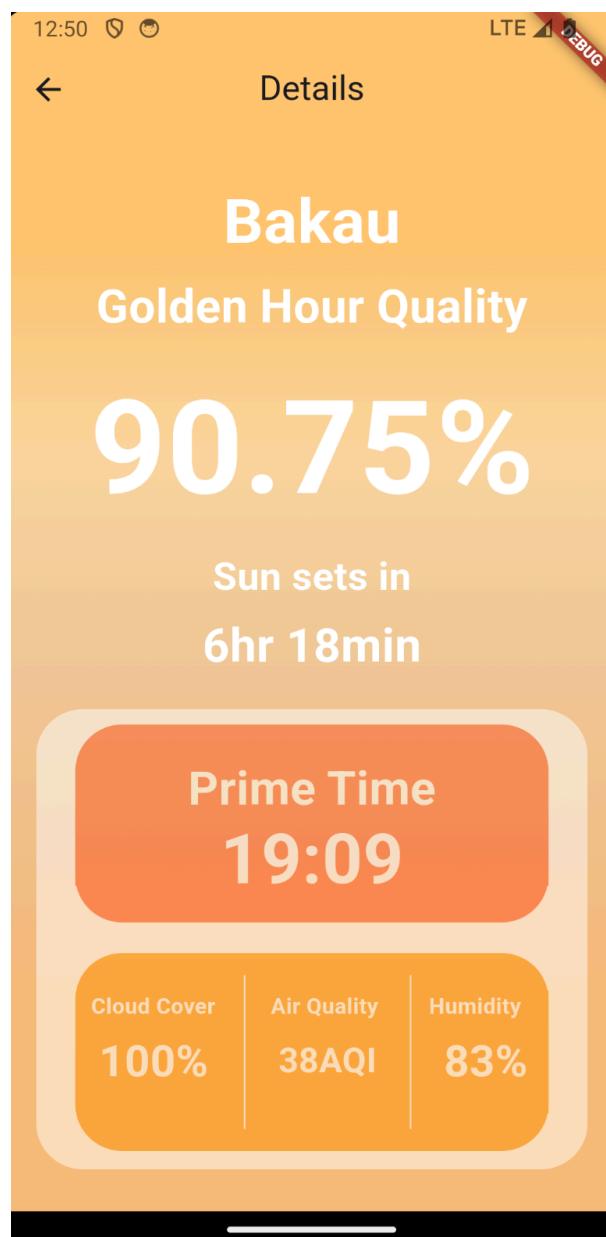
3.1.3 Home Page



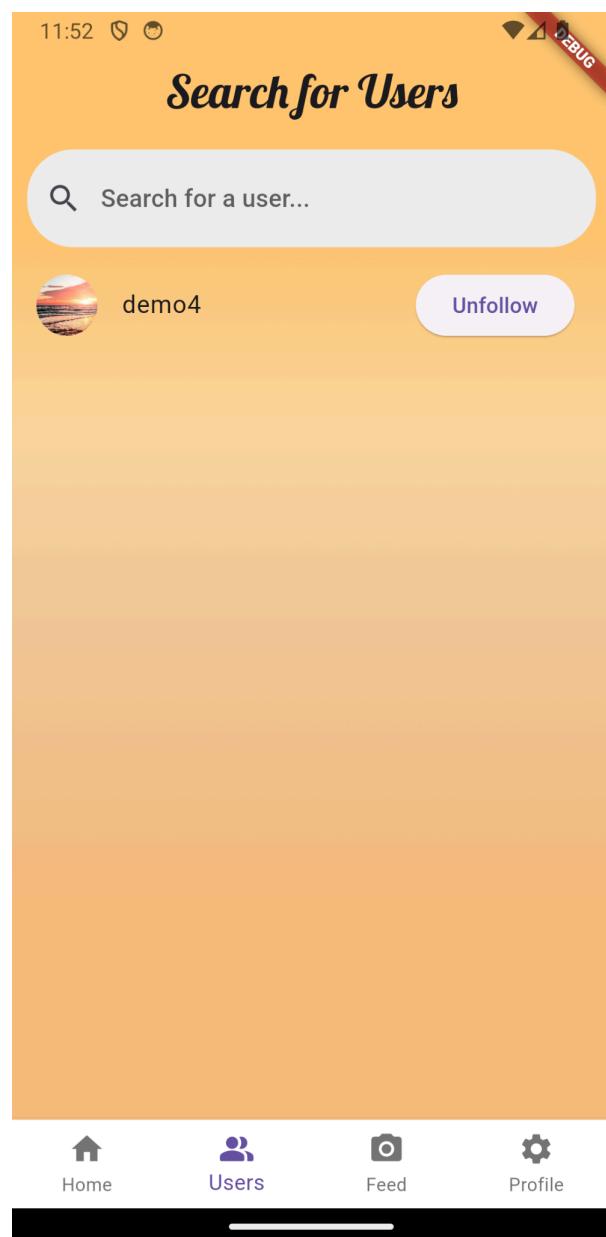
3.1.4 Sort Home Page



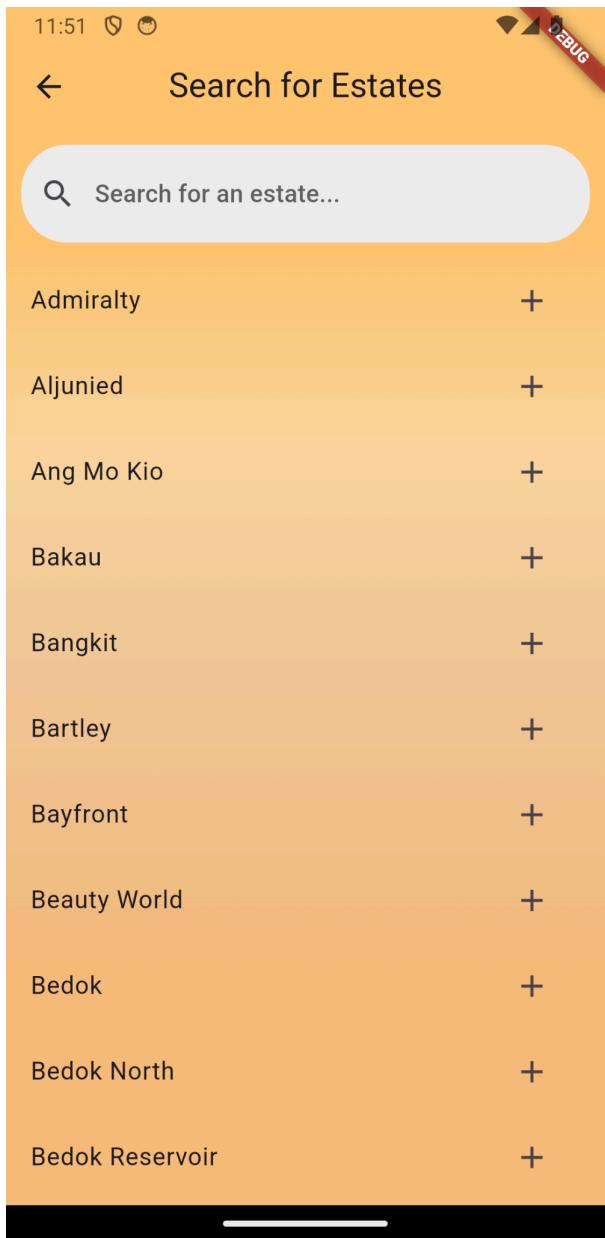
3.1.5 View Details



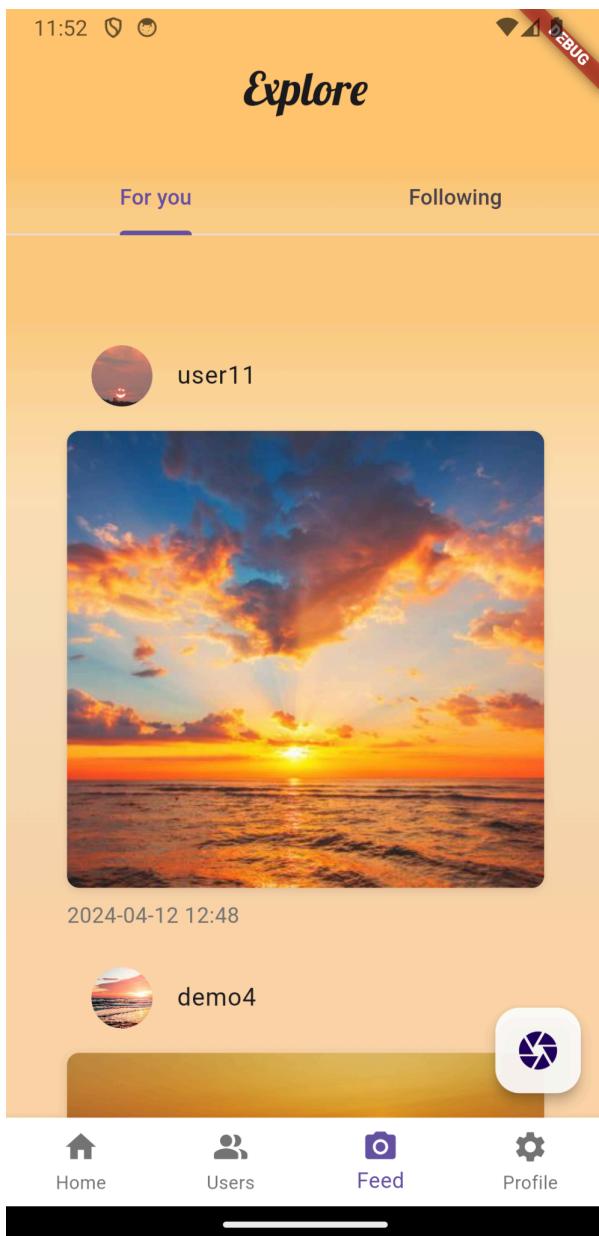
3.1.6 Search Users



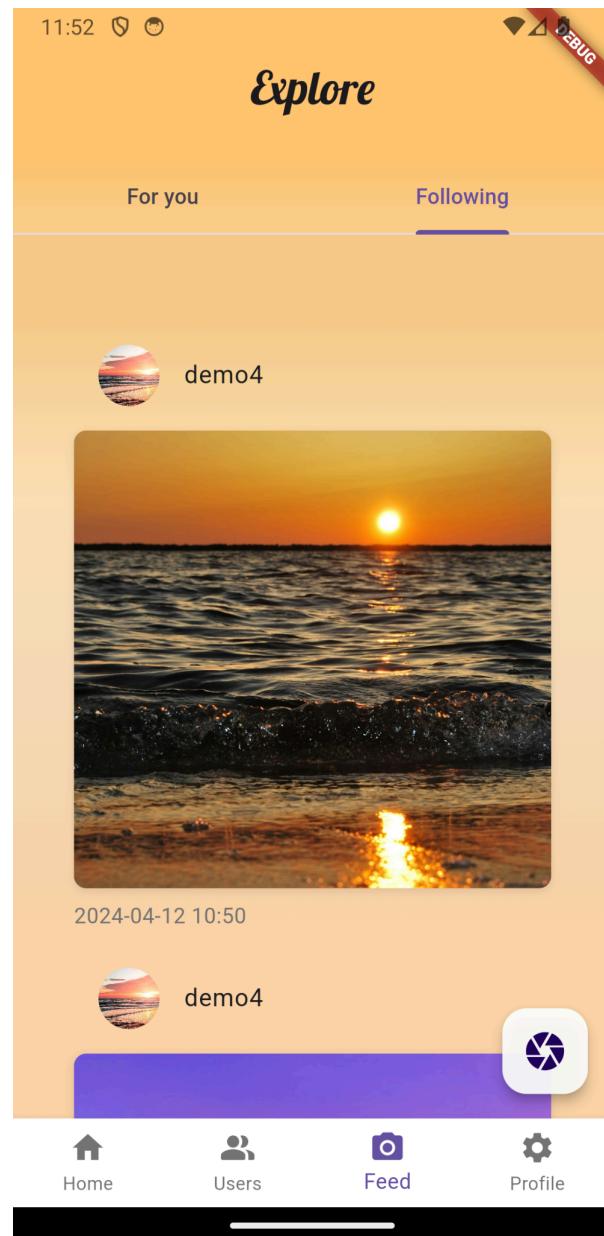
3.1.7 Search Location



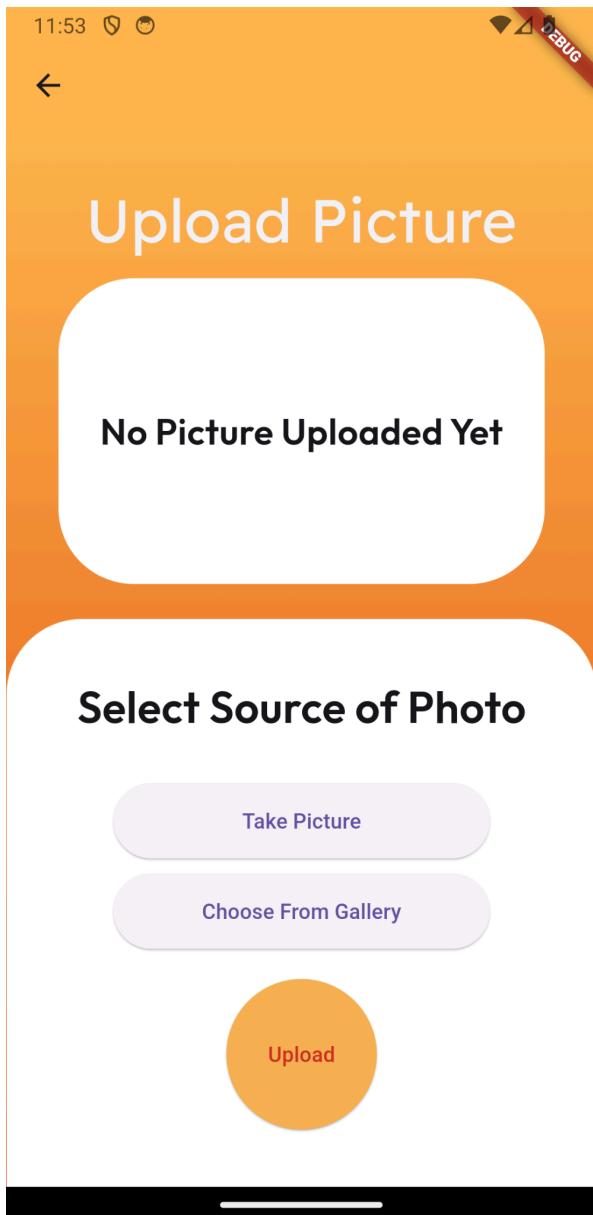
3.1.8 For You Feed



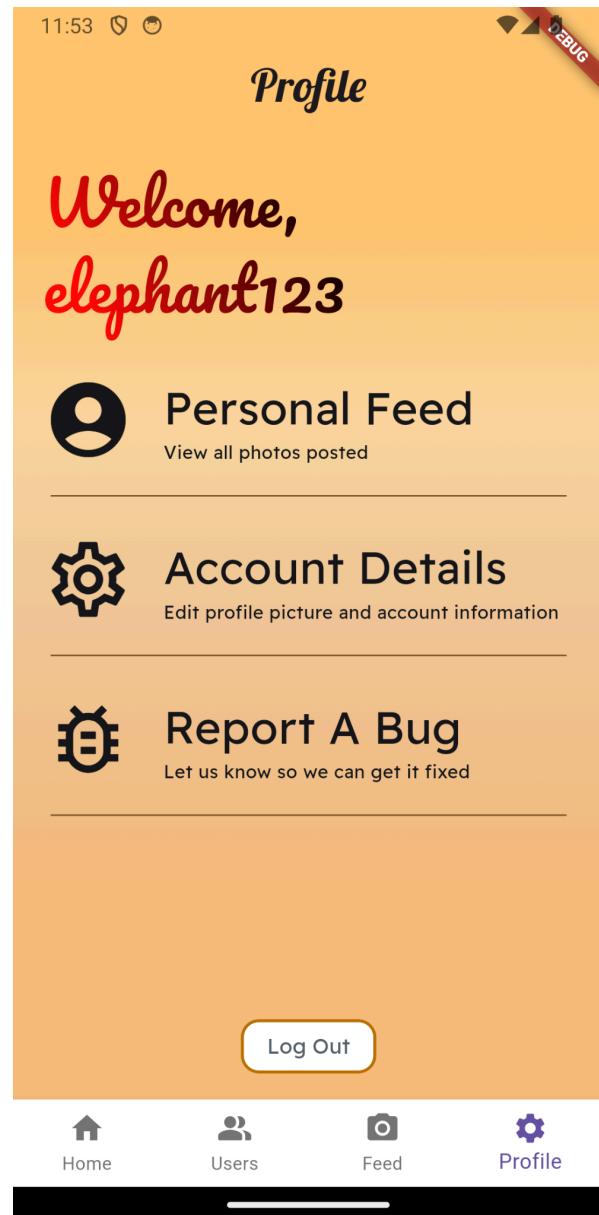
3.1.9 Following Feed



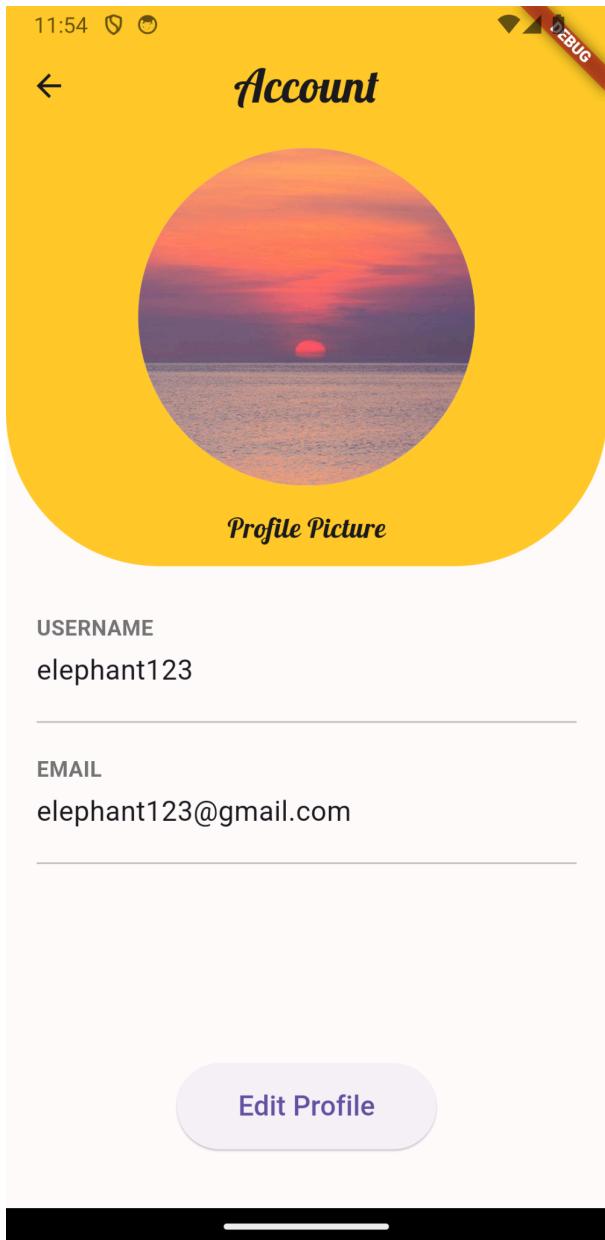
3.1.10 Upload Photo



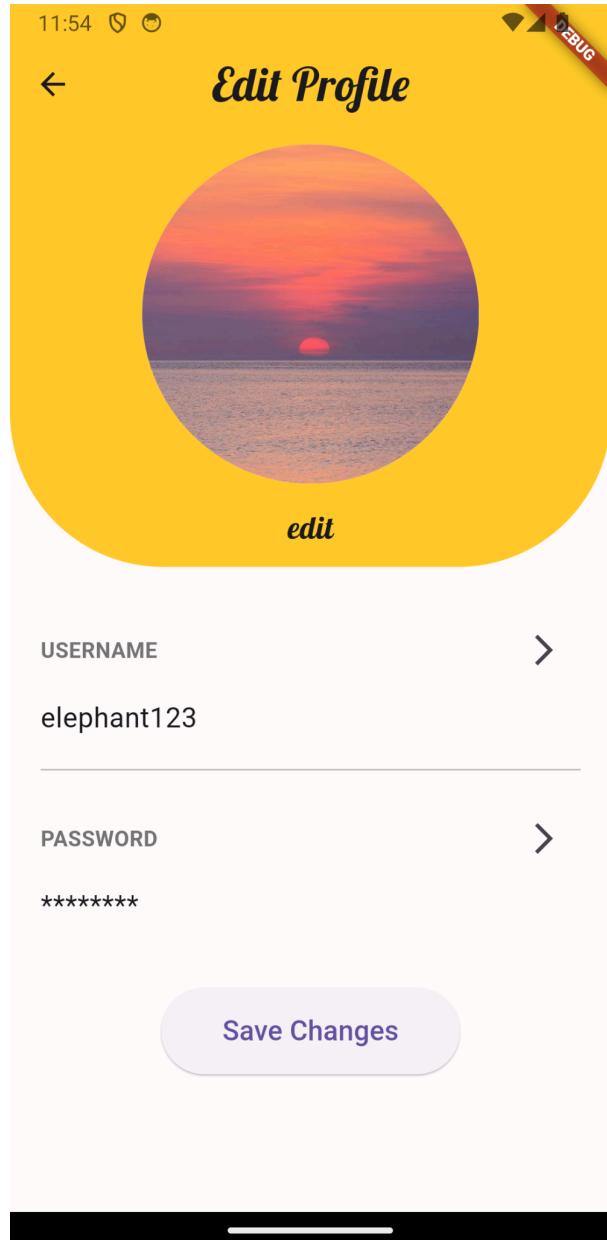
3.1.11 Profile



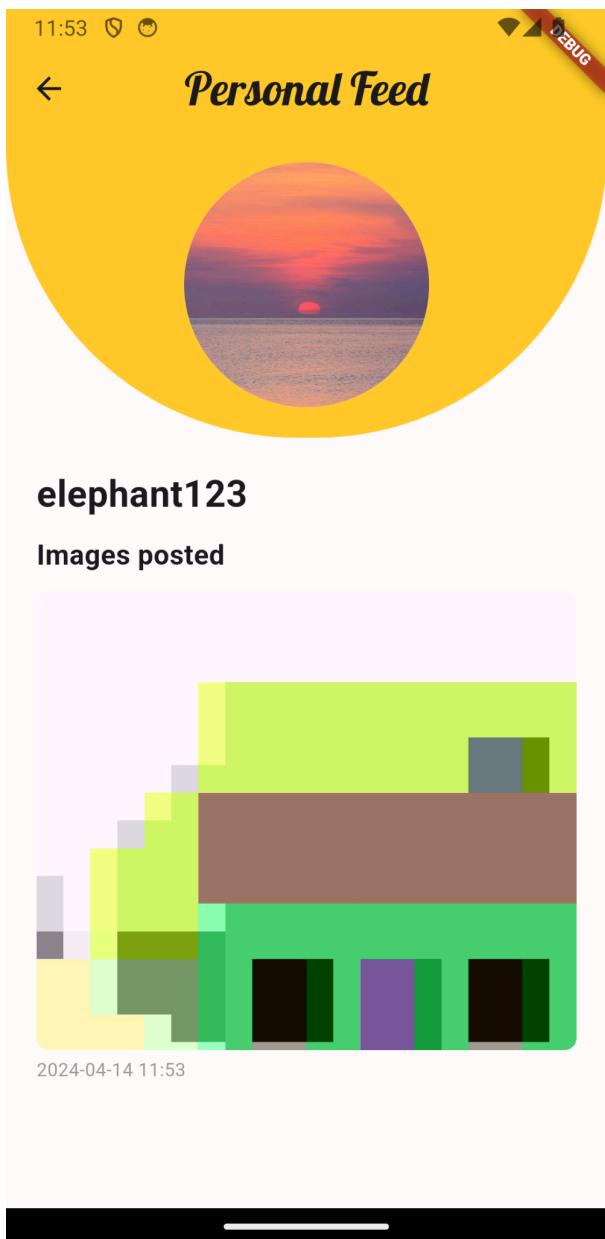
3.1.12 Account Details



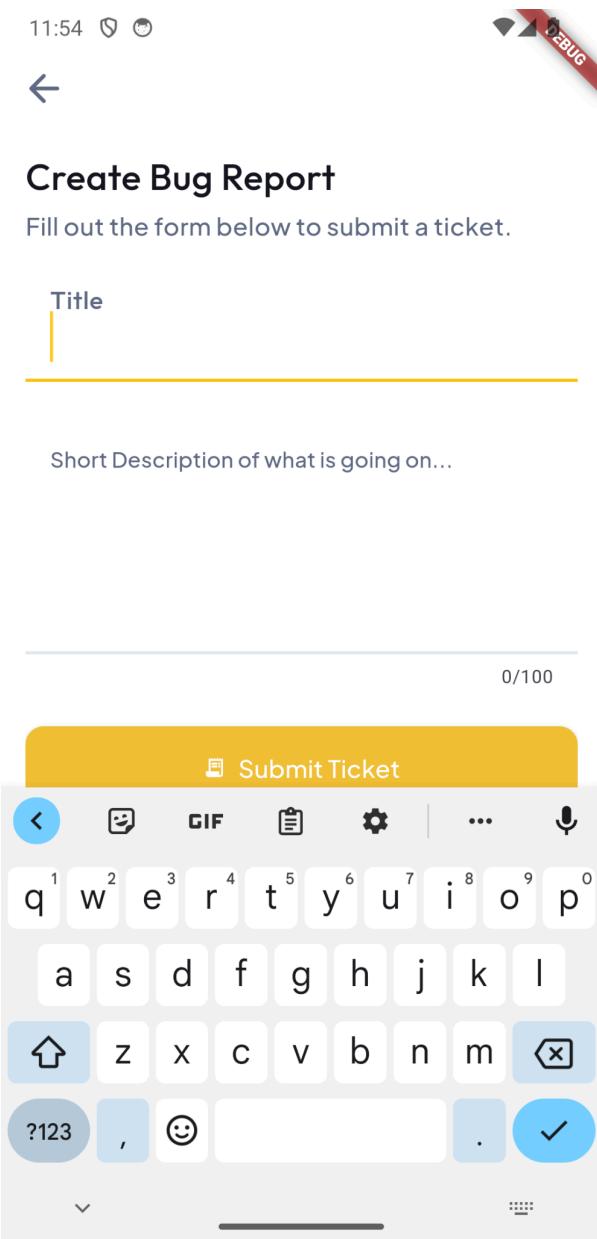
3.1.13 Edit Profile



3.1.14 Personal Feed



3.1.15 Report Bug



3.2 Hardware Interfaces

1. Skyscape is developed to run on ARM64 CPU and Android Version 14.0 and above.
2. Access to internet connection is required for Skyscape to run as expected due to the dependencies on external API calls and cloud-based database.

3.3 Software Interfaces

1. Front-end:
 - a. Flutter 3.19.2: Widget-centric design with extensive widget library and reactive framework for building user interfaces.
 - b. Dart 3.3.0: Programming language used to develop the Flutter project.
2. Back-end:
 - a. Firebase: Stores user related information for the application to be personalized.

3.4 Communications Interfaces

1. **HTTPS and REST API:** In order to retrieve meteorological information on specific coordinates and user data, HTTPS and REST API have been used.

4. System Features

4.1. Use case of Registration

Use Case ID:	REG1		
Use Case Name:	Registration		
Created By:	Li Yaohao	Last Updated By:	Kiersten Yeo
Date Created:	29/01/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	First time users must register for a new account. Having an account allows them to store pictures taken and share them online to various social media platforms.
Preconditions:	Nil
Postconditions:	The user has successfully created an account and logged in.
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the link to register. 2. Users are prompted to enter a username, email, password and confirm the password. 3. User submits required information. 4. System validates information, and stores login credentials into the database. 5. Users will be logged into the account.
Alternative Flows:	<p>REG1-AF-S1 If username is already registered:</p> <ol style="list-style-type: none"> 1. The system shows the error message “Username is already taken. Please choose a different username.” 2. The system will return to step 2. <p>REG1-AF-S2 If password does not meet requirements:</p> <ol style="list-style-type: none"> 1. The system shows the error message “Password must be at least 8 characters long and include at least one capital letter and one number.” 2. The system will return to step 2. <p>REG1-AF-S3 If re-entered password does not match the password:</p> <ol style="list-style-type: none"> 1. The system shows the error message “Passwords do not match.” 2. The system will return to step 2. <p>REG1-AF-S4 If email is invalid or already registered:</p> <ol style="list-style-type: none"> 3. The system shows the error message “Please supply a valid email.” 4. The system will return to step 2.
Exceptions:	EX1 The system is currently offline.
Includes:	Nil
Special Requirements:	Nil

Assumptions:	The user is connected to the internet.
Notes and Issues:	Nil

4.1.1 Functional Requirements

1. The system must have a register function on the login page for first time users.
 - 1.1. The registration form must include text fields for essential user information.
 - 1.1.1. The form must consist of username.
 - 1.1.2. The form must consist of email.
 - 1.1.3. The form must consist of a password.
 - 1.1.4. The form must consist of an additional password for verification.
 - 1.2. The system must verify that all fields are filled when the user presses “Submit”.
 - 1.3. The system must verify all fields after the user presses “Submit”.
 - 1.3.1. The username must be unique from all registered usernames.
 - 1.3.2. The email must be unique from all registered emails.
 - 1.3.3. The password must contain at least 1 capital letter, 1 number and have at least 8 characters.
 - 1.3.4. Both passwords must be exactly the same.
 - 1.3.5. The system must prompt the user on any errors in the above fields.
 - 1.4. The system must authenticate the given email address by the user.
 - 1.4.1. The email must be in the format of XXX@XXX.XXX
 - 1.5. The system must create an account for the user upon verification.
 - 1.6. The system must log the user into the homepage of the application.

4.2. Use case of Login

Use Case ID:	LOG1		
Use Case Name:	Login		
Created By:	Lim Zhi Yong	Last Updated By:	Kiersten Yeo
Date Created:	29/01/24	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	Users can login with their login details to access their recently-viewed history and starred locations.
Preconditions:	User already has a registered account.
Postconditions:	Users will login and the homepage will be shown.
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. Users will input their registered email and password into the system. 2. Users will click the login button. 3. The system will verify the email and the password to ensure that it is a valid account. 4. The user will be logged into the system and the homepage will be displayed.
Alternative Flows:	<p>LOG1-AF-S1 If the email does not exist in the database.</p> <ol style="list-style-type: none"> 1. Show error message “Incorrect email or password.” 2. The system will return to step 1. <p>LOG1-AF-S2 If the password does not match the correct password saved in the database.</p> <ol style="list-style-type: none"> 1. Show error message “Incorrect email or password.” 2. The system will return to step 1. <p>LOG1-AF-S3 If the user has previously logged in recently.</p> <ol style="list-style-type: none"> 1. This page will be skipped and upon starting the application, the user will be brought straight to the homepage without the need to login again.
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.2.1 Functional Requirements

1. The user must log into the system whenever the system restarts.
 - 1.1. The login page must display text fields for essential user information.
 - 1.1.1. The text field must consist of an email address.
 - 1.1.2. The test field must consist of a password.
 - 1.2. The system must verify that all fields are filled when the user presses “Login”.
 - 1.3. The system must verify all fields after the user presses “Login”.
 - 1.3.1. The email must match the email of an existing account in the database.
 - 1.3.2. The password must match the password of the account in the database.
 - 1.3.3. The system must prompt the user on any errors in the above fields.
 - 1.4. The system must log the user into the homepage of the application.

4.3. Use case of Access Homepage

Use Case ID:	HOME1		
Use Case Name:	Access Homepage		
Created By:	Li Yaohao	Last Updated By:	Kiersten Yeo
Date Created:	30/01/24	Date Last Updated:	14/04/24

Actor:	User
Description:	The homepage shows the quality of golden hour sorted in descending order for a day.
Preconditions:	User already has a registered account and has logged in.
Postconditions:	Nil
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. Default: Users can view the quality of sunsets sorted according to when the location is added to the homepage. 2. Users can click the top left button at the app bar to sort the location list based on descending quality of sunsets or alphabetical order. 3. Users can click on the date tab at the top of the screen to view the sunset qualities of a different date. 4. User can add locations by pressing the “+” button on the top right of the screen. 5. User can remove locations by swiping left on the locations. 6. Users can access other features from the homepage. <ol style="list-style-type: none"> 4.1 Clicking on the three other tabs in the bottom navigation bar ‘Users’, ‘Feed’, ‘Profile’ to bring the user to those pages respectively. 4.2 Clicking on a location will show the users the detailed information about that location.
Alternative Flows:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	The API calls are successful.
Notes and Issues:	Nil

4.3.1 Functional Requirements

1. The user must be able to view the details of the locations he/she has saved.
 - 1.1. The homepage must display the sunset quality in percentage.
 - 1.2. The user must be able to click on each location to view detailed information about each location.
2. The user must be able to sort the entries in the homepage.
 - 2.1. The user must be able to sort based on descending order of sunset quality.
 - 2.2. The user must be able to sort based on alphabetical order.
3. The user must be able to view the sunset qualities of the locations for different dates.

- 3.1. The user must be able to click on the dates in the date tab to view the sunset qualities for different dates.
4. The user must be able to add a location to the homepage by pressing the + icon at the top right hand corner of the homepage.
 5. The user must be able to delete a location currently in the homepage by swiping left on the location he wants to remove.
 6. The user must be able to access other features from the homepage.
 - 6.1. Clicking on the other tabs in the bottom navigation bar will bring the users to the respective pages.

4.4. Use case of Edit Homepage

Use Case ID:	FAV1		
Use Case Name:	Edit Homepage		
Created By:	Yoon Sang Won	Last Updated By:	Yoon Sang Won
Date Created:	31/01/24	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	User can edit the homepage by adding or removing locations which user searches
Preconditions:	<ol style="list-style-type: none"> 1. User has pressed the “+” button at the top right corner of the screen at the appbar 2. User has swiped left on any of the locations in the home page.
Postconditions:	Homepage will be updated with the locations the user has added or removed.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. User can add locations by pressing “+” button on the top right of the screen <ol style="list-style-type: none"> 1.1 User can search a location he wants in the search bar. 1.1.2 If the input location is in the list, it will show up on the screen. 1.2 User can also scroll through the list of locations to find the location he wants. 1.3 User can add this location to the homepage by pressing the + icon next to the location name. 1.4 Once successfully added, there will be a text at the bottom of the screen, “Location successfully added” to inform user adding was successful. <ol style="list-style-type: none"> 1.4.1 Once a location is added, user will be redirected back to homepage to view the new added location. 2. User can remove locations by swiping left on the locations. <ol style="list-style-type: none"> 2.1 User can choose to remove all locations. 2.2 Once swiped, the selected locations will be deleted from the homepage immediately and updated in the database.
Alternative Flows:	FAV1-AF-S1 If the location entered by the user is not in the list <ol style="list-style-type: none"> 1. The system will show a text saying “No results found” and user is not allowed to proceed until user enters a valid name.

	FAV1-AF-S2 If the location is already added to the homepage 1. If user tries to add a location that has already been added, the system will show a text saying “Location is already saved”
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.4.1 Functional Requirements

1. The user must be able to add a location to the homepage by pressing the + icon at the top right hand corner of the homepage.
 - 1.1. The user must be able to find the location he wants to add to the homepage.
 - 1.1.1. The user must be able to search for the desired location by entering the location in the search bar.
 - 1.1.1.1. The system must verify that the location is valid.
 - 1.1.2. The user must be able to scroll through the list of locations to find the desired location.
 - 1.2. The user must be able to add the location he found to the homepage by pressing the + icon on the right side of the location name.
 - 1.2.1. The system must show a message denoting a successful addition of location to the homepage.
 - 1.2.2. The system must redirect the user back to the homepage after a successful addition of location.
 - 1.3. The user must not be able to add the same location twice.
 - 1.3.1. The system must show an error message denoting that the location has already been added.
2. The user must be able to delete a location currently in the homepage by swiping left on the location he wants to remove.
 - 2.1. The system must show a message denoting a successful deletion of the location.

4.5. Use case of Calendar Function

Use Case ID:	CAL1		
Use Case Name:	Calendar function		
Created By:	Jin Ying	Last Updated By:	Yoon Sang Won
Date Created:	04/02/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	A calendar like function at the top of the homepage for users to change dates to view predicted quality of sunset.
Preconditions:	Another date has been pressed by the user on the date tab
Postconditions:	Data shown on the homepage is changed to the user's choice of date
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on one of the dates shown on the date tab at the top of the screen. 2. Date shown changes to the date chosen by the user. 3. Homepage will be refreshed, and the sunset quality of the locations will be changed to reflect that of the new date.
Alternative Flows:	<p>CAL1-AF-S1 If the user clicks on a date that is more than 7 days away</p> <ol style="list-style-type: none"> 1. System displays "No predictions for the selected date yet!" 2. System returns to step 2. <p>CAL1-AF-S2 If the user clicks on past dates</p> <ol style="list-style-type: none"> 1. System displays "Information expired." 2. System returns to step 2.
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	The API calls are successful.
Notes and Issues:	Nil

4.5.1 Functional Requirements

1. The system must have a calendar tab. (Weather forecasts are useful up to 7 days)
 - 1.1. The calendar tab must be displayed in weeks.
 - 1.2. The user must be able to click on different dates on the calendar to view details of different days.
 - 1.2.1. The user must be able to scroll sideways on the calendar tab to access other weeks.
 - 1.2.2. The user must not be able to access information that is more than 7 days away from the current date.
 - 1.2.2.1. The system must display an error message showing that prediction cannot be made for a date more than 7 days away.
 - 1.3. The system must show the sunset qualities based on the date that the user changed to.
 - 1.4. The system must refresh the homepage every time the user changes the date.

4.6. Use case of Location Details

Use Case ID:	DET1		
Use Case Name:	Location details		
Created By:	Yoon Sang Won	Last Updated By:	Yoon Sang Won
Date Created:	04/02/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	User is able to view the details of the sunset as well as other weather conditions that affect the sunset.
Preconditions:	User taps on one of the locations in the home page.
Postconditions:	User clicks the back button at the top left of the app bar to return back to the homepage
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. User taps on one of the locations added to his homepage. 2. System will display the detailed information regarding the sunset. These details include: <ol style="list-style-type: none"> 2.1: Location name 2.2: Sunset quality percentage 2.3: Exact time of sunset on the current day in the format HH:MM PM 2.4: Time left till sunset in hours and minutes 2.5: Cloud cover percentage 2.6: Humidity level in percentage 2.7: Air quality in AQI
Alternative Flows:	Nil
Exceptions:	EX1 No location has been added to the homepage
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.6.1 Functional Requirements

1. The user must be able to click on the locations list to view more details about that particular location.
 - 1.1. The detailed location page must display all the relevant information.
 - 1.1.1. This includes the location name.
 - 1.1.2. This includes Golden Hour quality.
 - 1.1.3. This includes sunset time in the format HH:MM PM.
 - 1.1.4. This includes time left to sunset in hours and minutes.
 - 1.1.5. This includes cloud cover percentage.
 - 1.1.6. This includes humidity level.

- 1.1.7. This includes air quality.
- 1.2. The displayed information must be fetched successfully from the API.
- 1.2.1. The information should be fetched every five minutes to ensure accuracy and responsiveness of the application.

4.7. Use case of Sort Locations

Use Case ID:	SORT1		
Use Case Name:	Sort locations		
Created By:	Yoon Sang Won	Last Updated By:	Yoon Sang Won
Date Created:	04/02/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	User can sort the locations in his homepage according to alphabetical order of the location name or the quality of sunset.
Preconditions:	User has entered the homepage.
Postconditions:	Order of the locations in the homepage will be sorted according to the alphabetical order of the location name or quality that user chooses.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> User taps the sort button at the top left hand corner of the app bar on the homepage. User can choose to sort the list by either quality of sunset in descending order or the alphabetical order of the location name. <ol style="list-style-type: none"> If the user sorts by quality, the list will be sorted starting from the location with the highest sunset quality. If the user sorts by the alphabetical order of the location name, the locations will be arranged in the alphabetical order of A to Z.
Alternative Flows:	SORT1-AF-S1 If there is no location in the homepage <ol style="list-style-type: none"> System will show a text saying "No location to be sorted!".
Exceptions:	EX1 If there is a tie between the quality <ol style="list-style-type: none"> The tie will be broken by alphabetical order.
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.7.1 Functional Requirements

1. The user must be able to sort the list of locations saved either by quality of sunset or by alphabetical order by pressing the sort button on the top left hand corner of the homepage.

- 1.1. The system must provide users a button to choose whether they want the locations to be sorted by quality of sunset or by alphabetical order.
 - 1.1.1. The user must be able to sort the location list in the order of quality by pressing the “Quality” button.
 - 1.1.1.1. The system must calculate the golden hour quality percentage for all the locations in the list.
 - 1.1.1.2. The system must sort the locations in the order of location with highest quality to lowest quality.
 - 1.1.2. The user must be able to sort the location list in the alphabetical order by pressing the “Alphabet” button.
 - 1.1.2.1. The system must sort the locations in the order of locations starting from “A” to “Z”.
- 1.2. The system must recognise if there are no saved locations and inform the user to save at least 2 locations first before sorting them.

4.8. Use case of Upload Pictures

Use Case ID:	POST1		
Use Case Name:	Upload Pictures		
Created By:	Lim Zhi Yong	Last Updated By:	Kiersten Yeo
Date Created:	06/02/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	Users can upload photos onto the feed which will be visible to everyone.
Preconditions:	User has navigated to the feedpage and entered the uploading picture page.
Postconditions:	Upon successful uploading, a message will be displayed at the bottom of the screen above the navigation bar, with the text “Picture uploaded successfully”.
Priority:	Low
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Upon entering the feed page on the navigation bar located at the bottom of the screen, the user will click onto the camera button to upload pictures. 2. User will get to select either the “Take Picture” option or the “Choose From Gallery” option. <ol style="list-style-type: none"> 2.1 If the user selects the “Take Picture” option, the phone’s camera will be turned on and the user will be able to take a photo which will be uploaded. 2.2 If the user selects the “Choose From Gallery” option, the gallery application will open and the user will be able to select a picture to upload. 3. User will see a preview of the photo before deciding if he/she would like to confirm the upload. 4. User can click on the “Upload Picture” option to confirm the upload. <ol style="list-style-type: none"> 4.1 Upon successful upload, a message will be displayed at the bottom of the screen with the text “Picture uploaded successfully”.

Alternative Flows:	POST1-AF-S1 If the user clicks on back 1. The system will return to the profile page.
Exceptions:	EX1 If the user clicks on “Upload Picture” before uploading one 1. The system will show an error message “You have not yet selected a picture”.
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.8.1 Functional Requirements

1. The system must allow the user to upload pictures of sunsets onto the feed.
 - 1.1. The system must have a secure and reliable database to store photos uploaded by the user.
 - 1.2. The system must provide the users with a method to upload pictures.
 - 1.2.1. The system must provide the option of choosing previously taken pictures of sunsets from the phone’s gallery.
 - 1.2.1.1 The system must request permission from the user to access the user’s photo gallery.
 - 1.2.2. The system must also provide the option of taking the picture of the sunset directly with the camera.
 - 1.2.2.1 The system must request permission from the user to access the user’s phone camera.
 - 1.2.3. After a photo is chosen, the system must ask the user to confirm the picture before uploading it.

4.9. Use case of Follow User

Use Case ID:	FOL1		
Use Case Name:	Follow User		
Created By:	Lim Zhi Yong	Last Updated By:	Kiersten Yeo
Date Created:	06/02/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	User can search for another user and follow them.
Preconditions:	The other user has created an account.
Postconditions:	The other user’s username appears in the current user’s user tab.
Priority:	Low

Frequency of Use:	Medium
Flow of Events:	<p>1. Upon navigating to the Users bar located at the bottom of the screen, the user will be able to search for other users in the search bar to follow.</p> <p>2. User will get to input the username of the other user in the search bar.</p> <p>3. User has to click the “Follow” button to follow another user. After clicking “Follow”, the “Follow” button will change to “Unfollow” to indicate that the current user has successfully followed another user.</p> <p>4. The list of followed users will be displayed below the search bar.</p>
Alternative Flows:	<p>FOL1-AF-S1 If the user inputs an invalid username</p> <ol style="list-style-type: none"> 1. The system will show an error message “No such user found”. 2. User can re-enter a different username to find different users. <p>POST1-AF-S2 If the user clicks on “X” to cancel searching</p> <ol style="list-style-type: none"> 1. The system will return to the default users bar that displays the list of existing followed users.
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.9.1 Functional Requirements

1. The user must be able to search for and follow other users.
 - 1.1. The system must have a reliable database to store all the different users' data.
 - 1.1.1. The system must provide a search bar for the user to search for other usernames.
 - 1.1.2. The system must be able to search the database for different users based on the username inputted into the search bar.
 - 1.1.3. The system must recognise and store the new connection when the user follows another user.
2. The user must be able to view the list of users he/she is following below the search bar in the User tab.
 - 2.1. The system must display all the other users the user is connected to.
3. The user must be able to view the uploaded pictures of the users he/she is following in the Following tab of the Feed page.

4.10. Use case of View and Edit Following

Use Case ID:	FOL1		
Use Case Name:	View and Edit Following		
Created By:	Lim Zhi Yong	Last Updated By:	Kiersten Yeo
Date Created:	08/02/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	User can view and edit the accounts that he/she is following
Preconditions:	Nil
Postconditions:	Nil
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. Upon navigating to the Users tab, the user is able to view the profile of the users he/she follows. 2. The user will click the “Unfollow” button beside the username of a user to unfollow the user.
	ACC1-AF-S1 If the user clicks on back: <ol style="list-style-type: none"> 1. The system will return to the Profile page.
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.10.1 Functional Requirements

1. The user must be able to view the list of users he/she is following below the search bar in the User tab.
 - 1.1. The system must display all the other users the user is following.
2. The user must be able to unfollow a user that he/she is currently following.
 - 2.1. The system provides users with an “Unfollow” button for every user currently being followed.
 - 2.2. The system must recognise and remove the old connection when the user unfollows another user.

4.11. Use case of Account Settings

Use Case ID:	ACC1		
Use Case Name:	Edit Account Details		
Created By:	Lim Zhi Yong	Last Updated By:	Yoon Sang Won
Date Created:	08/02/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	User can edit username, profile picture and password.
Preconditions:	User must be logged in
Postconditions:	User personal details are changed.
Priority:	Medium
Frequency of Use:	Low
Flow of Events:	<p>1. Upon entering the profile page by clicking onto the profile button on the navigation bar located at the bottom of the screen, the user will click onto the “Account Settings” tab.</p> <p>2. Once inside the “Account Settings” tab, the user is able to view the current account details, which are</p> <ol style="list-style-type: none"> 1. Profile Picture 2. Username 3. Email <p>3. The user will click onto the “Edit Profile” button that is located below the profile picture in order to make changes</p> <p>4. Upon entering the “Edit Profile” page, users can make changes to the following fields</p> <ol style="list-style-type: none"> 1. User can edit the profile picture and will get to select either the “Take Picture” option or the “Choose From Gallery” option. <ul style="list-style-type: none"> 4.1.1. If the user selects the “Take Picture” option, the phone’s camera will be turned on and the user will be able to take a photo which will be uploaded. 4.1.2. If the user selects the “Choose From Gallery” option, the gallery application will open and the user will be able to select a picture to upload. 4.1.3. User will see a preview of the photo before deciding if he/she would like to confirm the upload. 4.1.4. User can click on the “Upload Picture” option to confirm the upload. 4.1.5. The system will return to the profile page. 2. User can edit the username. <ul style="list-style-type: none"> 4.2.1. The user can input a new username into the username field text. 3. User can edit the password. <ul style="list-style-type: none"> 4.2.1. The user can input a new password into the password field text. 4.2.2. The user needs to verify password by keying in the same password again. 5. User can click “Save Changes” button once everything has been confirmed.
	ACC1-AF-S1 If the user clicks on back:

	<p>1. The system will return to the Profile page.</p> <p>ACC1-AF-S2 If the user clicks on back when he is in the Edit Picture page:</p> <ol style="list-style-type: none"> 1. The system will return to the Edit Profile page. <p>ACC1-AF-S3 If the new username is not unique:</p> <ol style="list-style-type: none"> 1. The system shows the error message “Please enter another username”. <p>ACC1-AF-S4 If password does not meet requirements:</p> <ol style="list-style-type: none"> 1. The system shows the error message “Password does not meet requirements. Please enter a different password”. <p>ACC1-AF-S5 If password keyed into “confirm password” tab and “new password” tabs are different:</p> <ol style="list-style-type: none"> 1. The system shows the error message “Passwords do not match”.
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Email cannot and should not be edited.

4.11.1 Functional Requirements

1. The user must be able to access the account page.
 - 1.1. The account page must display relevant user information.
 - 1.1.1. The account page must display the user profile picture.
 - 1.1.2. The account page must display the username.
 - 1.1.3. The account page must display the user email.
 - 1.2. The user must be able to change account details.
 - 1.2.1. The user must be able to change the profile picture.
 - 1.2.1.1. The system must display a button to allow the user to change the profile picture.
 - 1.2.1.1.1. The user must be able to choose between taking a photo and selecting a photo from his picture gallery
 - 1.2.1.1.1.1. The system must open the camera function if the user chooses to take a photo.
 - 1.2.1.1.1.2. The system must open the picture gallery if the user chooses to select the photo from his gallery.
 - 1.2.1.1.2. The system must change the profile picture to the selected photo.
 - 1.2.2. The user must be able to change the username.
 - 1.2.2.1. The system must show a text field for a new username.
 - 1.2.3. The user must be able to change the password.
 - 1.2.3.1. The system must show 2 text fields for a new password (one for verification).
 - 1.2.4. The system must verify all fields after the user presses “Save Changes”.

- 1.2.4.1. The username must be unique from existing users.
- 1.2.4.2. The password must contain at least 1 capital, 1 number and have at least 8 characters.
- 1.2.4.3. Both passwords in “New password” tab and “Confirm password” tab must be exactly the same.
- 1.2.4.4. The system must prompt the user on any errors in the above fields.

4.12. Use case of Report Bug

Use Case ID:	REP1		
Use Case Name:	Report Bug		
Created By:	Lim Zhi Yong	Last Updated By:	Yoon Sang Won
Date Created:	06/02/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	User can report a bug that he/she has found within the application.
Preconditions:	A bug has been found in the application.
Postconditions:	Nil
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> Upon entering the profile page by clicking onto the profile button on the navigation bar located at the bottom of the screen, the user will click onto the “Report a Bug” button. The user will be provided a field for the title and a field for the description of the bug. The user can type up to 100 characters under description. The user can click on “Submit Ticket” to submit the report. The system will return to the profile page.
Alternative Flows:	POST1-AF-S1 If the user clicks on back <ol style="list-style-type: none"> The system will return to the profile page.
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.12.1 Functional Requirements

- The user must be able to report a bug, if any, that he/she encountered while using the application.
 - The system must provide users a way to report bugs found.

- 1.1.1. This is to be located in the profile page, “Report a bug” function.
- 1.1.2. The user must be given a text box to write a title for the bug report.
- 1.1.3. The user must be given a text box to describe the type of bugs he encountered.
 - 1.1.3.1. The system must allow the description field to be maximum 100 characters long.
- 1.1.4. The system must ensure that the user submits a valid bug encounter report.
 - 1.1.4.1. This can be done by ensuring that the title text field and the description text field filled by the user is not blank.

4.13. Use case of View All Photos Uploaded

Use Case ID:	VALL1		
Use Case Name:	View All		
Created By:	Li Yaohao	Last Updated By:	Yoon Sang Won
Date Created:	28/03/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	User can view picture of sunsets posted by all the other users using the application
Preconditions:	A user must have an account registered and logged in
Postconditions:	Nil
Priority:	Low
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. Users navigate to the feed page from the bottom navigation bar. 2. The feed page contains two tabs, “All” and “Following”. The default tab is the “All” tab. 3. Users can scroll up and down the page to view sunset photos posted by other users. <ul style="list-style-type: none"> 3.1 Above the photos posted, the username and profile picture of the user who posted it is included. 3.2 Below the photos posted, the time stamp of the time the photo is uploaded is included. 4. Users can perform additional functions in the feed page, which are <ol style="list-style-type: none"> 4.1. Clicking on the username of other uses to access that user’s personal page. <ul style="list-style-type: none"> 4.1.1 The personal page shows all photos posted by that specific user. 4.2. Clicking on the camera button at the bottom right of the screen to access the uploading picture page. 4.3. Clicking on the “Following” tab at the top right of the screen, below the app bar, to access the Following tab.
Alternative Flows:	Nil
Exceptions:	Nil
Includes:	Nil

Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.13.1 Functional Requirements

1. The user must be able to view all the photos other users have posted in the “View All” tab.
 - 1.1. The user must be able to scroll vertically in the page to view and access all the photos other users have uploaded.
 - 1.2. The user must be able to view the information related to the photo uploaded.
 - 1.2.1. This includes the username of the user who uploaded the photo.
 - 1.2.1.1. This is shown above the photo that has been uploaded.
 - 1.2.2. This includes the time stamp of the time the photo was uploaded.
 - 1.2.2.1. This is shown below the photo that has been uploaded.
 - 1.3. The user must be able to click on a username to access that user’s personal page.
 - 1.3.1. The user must be able to see all the photos that the specific user has uploaded.
 - 1.4. The user must be able to click the camera icon on the bottom right hand corner of the page to upload photos to the feed.
 - 1.5. The user must be able to view photos of following users by clicking on the “Following” tab on the top right hand corner of the screen.

4.14. Use case of View Photos by Following

Use Case ID:	VFLW1		
Use Case Name:	View Photos by Following		
Created By:	Li Yaohao	Last Updated By:	Yoon Sang Won
Date Created:	28/03/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	User can view pictures of sunsets posted by the users he/she is currently following.
Preconditions:	<ol style="list-style-type: none"> 1. A user must have an account registered and logged in. 2. A user must have followed other users
Postconditions:	Nil
Priority:	Low
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. Users navigate to the feed page from the bottom navigation bar.

	<p>2. The feed page contains two tabs, “All” and “Following”. The default tab is the “All” tab.</p> <p>3. Users click on the “Following” tab to access the following page to view photos posted only by other users that the current user is following.</p> <p>3.1 Above the photos posted, the username and profile picture of the user who posted it is included.</p> <p>3.2 Below the photos posted, the time stamp of the time the photo is uploaded is included.</p> <p>4. Users can perform additional functions in the feed page, which are</p> <p>4.1. Clicking on the username of other users to access that user’s personal page.</p> <p>4.1.1 The personal page shows all photos posted by that specific user.</p> <p>4.2. Clicking on the camera button at the bottom right of the screen to access the uploading picture page.</p> <p>4.3. Clicking on the “View All” tab at the top right of the screen, below the app bar, to be brought back to the default feed page.</p>
Alternative Flows:	FPG2-AF-S1 If the user is not following anyone: 1. The page will display the text message “You are not following anyone. Please follow other users to view their photos”
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.14.1 Functional Requirements

1. The user must be able to view all the photos that the users who he/she follows have posted in the “Following” tab.
 - 1.1. The user must be able to scroll vertically in the page to view and access all the photos the users who he/she follows have uploaded.
 - 1.2. The user must be able to view the information related to the photo uploaded.
 - 1.2.1. This includes the username of the user who uploaded the photo.
 - 1.2.1.1. This is shown above the photo that has been uploaded.
 - 1.2.2. This includes the time stamp of the time the photo was uploaded.
 - 1.2.2.1. This is shown below the photo that has been uploaded.
 - 1.3. The user must be able to click on a username to access that user’s profile page.
 - 1.3.1. The user must be able to see all the photos that the specific user has uploaded.
 - 1.4. The user must be able to click the camera icon on the bottom right hand corner of the page to upload photos to the feed.
 - 1.5. The user must be able to view photos of all users by clicking on the “View All” tab on the top left hand corner of the screen.

- 1.6. The system must show an error message if the user is not following anyone.

4.15. Use case of Access other User's Profile

Use Case ID:	FOLW1		
Use Case Name:	Access other User's Profile		
Created By:	Li Yaohao	Last Updated By:	Kiersten Yeo
Date Created:	28/03/2024	Date Last Updated:	14/04/24

Actor:	User (initiating)
Description:	User can access the profile of other users
Preconditions:	Users must see the username of other users before being able to enter their profile. <ol style="list-style-type: none"> 1. Usernames of other users can be accessed from searching that user's username in the Users tab. 2. Usernames of other users can be found in the feed page, both in the "For You" and "Following" Tab.
Postconditions:	Users entered the profile of other users
Priority:	Low
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the username of other users. 2. User is directed to the profile page of that specific user to view all photos posted by that user.
Alternative Flows:	FOLW1-AF-S1 If the user has not uploaded any picture: 1.The page will only show profile picture of the user, with a text message "No photos have been posted by the user"
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.15.1 Functional Requirements

1. The user must be able to search for other users' profiles in the User tab.
 - 1.1. The user must be given a text box to input other users' usernames.
 - 1.2. The system must display the profile page of the user who has the username.
 - 1.2.1. The profile page displays the other user's profile picture and username.
 - 1.2.2. The user must be able to scroll vertically in the page to view and access all the photos the other user uploaded.
2. The system must show an error message if the username does not exist in the database.

4.16. Use case of Access Personal Profile

Use Case ID:	PPF1		
Use Case Name:	Access Personal Profile		
Created By:	Li Yaohao	Last Updated By:	Yoon Sang won
Date Created:	28/03/2024	Date Last Updated:	14/04/2024

Actor:	User (initiating)
Description:	Users can view their own profiles and see what they had posted.
Preconditions:	Users must have registered for an account and be logged in
Postconditions:	
Priority:	Low
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. User navigate to the profile page by clicking profile on the bottom navigation bar. 2. User can view all of the photos previously posted 3. Below the photos posted, the time stamp of the time the photo is uploaded is included.
Alternative Flows:	PPF1-AF-S1 If the user has not uploaded any picture: 1.The page will only show profile picture of the user, with a text message “No photos have been posted by you”
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

4.16.1 Functional Requirements

1. The user must be able to view his own profile by clicking on the “Personal Feed” button in the “Profile” tab in the bottom navigator.
 - 1.1. The user must be able to view all the photos he has posted in the profile.
 - 1.1.1. The user must be able to see the timestamp of the time the photo was uploaded.
1.1.1.1. This is shown below the photo that was uploaded.
 - 1.2. The system must show a message when no photo has been uploaded by the user.

5. Other Nonfunctional Requirements

Usability	<ul style="list-style-type: none">- The user interface must be intuitive and user-friendly in order to cater to users from different levels of technical expertise.
Reliability	<ul style="list-style-type: none">- After a system reboot, the full system functionalities must be restored within 2 minutes.
Performance	<ul style="list-style-type: none">- When a location is inputted into the system, the system must be able to provide accurate data within 5 seconds.- The time of sunset must be accurate to within +/- 5 minutes.- When a user input is detected, the app must respond within 3 seconds.
Supportability	<ul style="list-style-type: none">- The system must be able to function beyond Singapore when a different API is provided to it.
Security	<ul style="list-style-type: none">- The system should require authentication to be logged into their accounts.- The sensitive information should be encrypted before being stored in a database.

6. Other Requirements

6.1. Black Box Testing (highlight indicates the invalid input out of multiple inputs)

6.1.1. Login Functionalities

Test ID	Test Case Description	Input: Email	Input: Password	Expected Output	Actual Output	Pass/Fail
EC1	Verify successful login with valid email and password	sunsetlover72@gmail.com	SUNSETlover72	User is logged in successfully	User is logged in successfully	Pass
EC2	Verify error message when email is invalid	sunsetlover72	SUNSETlover72	Error message "Invalid email or password"	Error message "Invalid email or password"	Pass
EC3	Verify error message when password is invalid	sunsetlover72@gmail.com	sunsetlover72	Error message "Invalid email or password"	Error message "Invalid email or password"	Pass
EC4	Verify error message when email is unregistered	sunsetlover73@gmail.com	SUNSETlover73	Error message "Invalid email or password"	Error message "Invalid email or password"	Pass
EC5	Verify error message when email is empty		SUNSETlover72	Error message "Enter an email"	Error message "Enter an email"	Pass
EC6	Verify error message when password is empty	sunsetlover72@gmail.com		Error message "Enter a valid password"	Error message "Enter a valid password"	Pass

6.1.2. Register/Sign up Functionalities

Test ID	Test Case Description	Input: Username	Input: Email	Input: Password	Input: Confirm password	Expected Output	Actual Output	Pass/Fail
EC1	Verify successful signup with valid information	sunsetlover74	sunsetlover74@gm ail.com	SUNSETlover74	SUNSETlover74	User is signed up successfully and logged in	User is signed up successfully and logged in	Pass
EC2	Verify error message when username is already registered, while email is unregistered	sunsetlover72	sunsetlover75@gm ail.com	SUNSETlover75	SUNSETlover75	Error message "Username is already taken. Please choose a different username"	Error message "Username is already taken. Please choose a different username"	Pass
EC3	Verify error message when email is already registered, while username is unregistered	sunsetlover75	sunsetlover74@gm ail.com	SUNSETlover74	SUNSETlover74	Error message "Please supply a valid email"	Error message "Please supply a valid email"	Pass
EC4	Verify error message when password and confirm password do not match	sunsetlover75	sunsetlover75@gm ail.com	SUNSETlover75	sunsetlover75	Error message "Passwords do not match"	Error message "Passwords do not match"	Pass
EC5	Verify error message when email is invalid	sunsetlover75	sunsetlover75	SUNSETlover75	SUNSETlover75	Error message "Please supply a valid email"	Error message "Please supply a valid email"	Pass

EC6	Verify error message when password does not meet requirements	sunsetlover75	sunsetlover75@gmail.com	sunset	sunset	Error message "Password must be at least 8 characters long and include at least one capital letter and one number"	Error message "Password must be at least 8 characters long and include at least one capital letter and one number"	Pass
-----	---	---------------	-------------------------	--------	--------	--	--	------

6.1.3. Add location Functionalities

Test ID	Test Case Description	Input	Expected Output	Actual Output	Pass/Fail
EC1	Verify successful searching of location	Toa Payoh	Toa Payoh	Toa Payoh	Pass
EC2	Verify successful adding of location	Button press “+” next to location name (e.g. Toa Payoh)	Pop up message “Toa Payoh added to favourites”	Pop up message “Toa Payoh added to favourites”	Pass
EC3	Verify successful removal of location	Swipe left from saved location (e.g. Toa Payoh)	Pop up message “Toa Payoh removed”	Pop up message “Toa Payoh removed”	Pass
EC4	Verify error message when adding duplicate location	Button press “+” next to location that is already added (e.g. Toa Payoh)	Pop up message “Toa Payoh is already saved”	Pop up message “Toa Payoh is already saved”	Pass
EC5	Verify error message adding location not in database	Johor Bahru			Pass

6.1.4. Upload Feed Functionalities

Test ID	Test Case Description	Input	Expected Output	Actual Output	Pass/Fail
EC1	Verify successful uploading of picture (camera)	Button press "Take Picture", picture taken from camera, button press "Upload"	Pop up message "Image uploaded successfully"	Pop up message "Image uploaded successfully"	Pass
EC2	Verify successful uploading of picture (choose from gallery)	Button press "Choose from gallery", picture taken from files, button press "Upload"	Pop up message "Image uploaded successfully"	Pop up message "Image uploaded successfully"	Pass

6.1.5. Follow Users Functionalities

Test ID	Test Case Description	Input Username	Expected Output	Actual Output	Pass/Fail
EC1	Search for username that exists in the database and is currently not followed	"jy"	Searched user is shown and current user is able to follow that user	User is shown and a button exists to follow the user	Pass
EC2	Search for username that exists in the database and is currently followed	"sw"	Searched user is shown and current user is able to unfollow that user	User is shown and a button exists to unfollow the user	Pass
EC3	Search for username that does not exist in the database	"name"	Message shown to show that the user is not found	Message is displayed saying 'No user found'	Pass
EC4	Search for username that exists in the database in a different case (upper, lower cases) and the changed username does not exist in the database	"SW"	Message show to show that the user is not found	Message is displayed saying 'No user found'	Pass
EC5	Search with no inputs	""	Show a list of followed users and allow the current user to unfollow them	List of users who are followed are shown and buttons exist for each user to unfollow them	Pass

6.1.6. Edit Profile Functionalities

Test ID	Test Case Description	Input Information	Expected Output	Actual Output	Pass/Fail
EC1	Change username to a username that does not exist in the database	Username = "new123"	Current username changes to the selected username	Username changes to the selected username and a message shows saying 'Profile Updated'	Pass
EC2	Change username to a username that already exists in the database	Username = "sw"	Error message shows saying that username cannot be changed and username does not change	Error message is displayed saying that 'Username already taken' and username is not changed	Pass
EC3	Change password to a different password that meets the condition of having a length of more than 8, contains an uppercase letter, and has a number	Password = "Password123"	Current password changes to the selected password	Password changes to the selected password and a message shows saying 'Profile Updated'	Pass
EC4	Change password to a different password that is shorter than 8 characters but has an uppercase letter and a number	Password = "Pass1"	Error message shows saying that password cannot be changed and password does not change	Error message is displayed saying that 'Password is invalid' and password is not changed	Pass

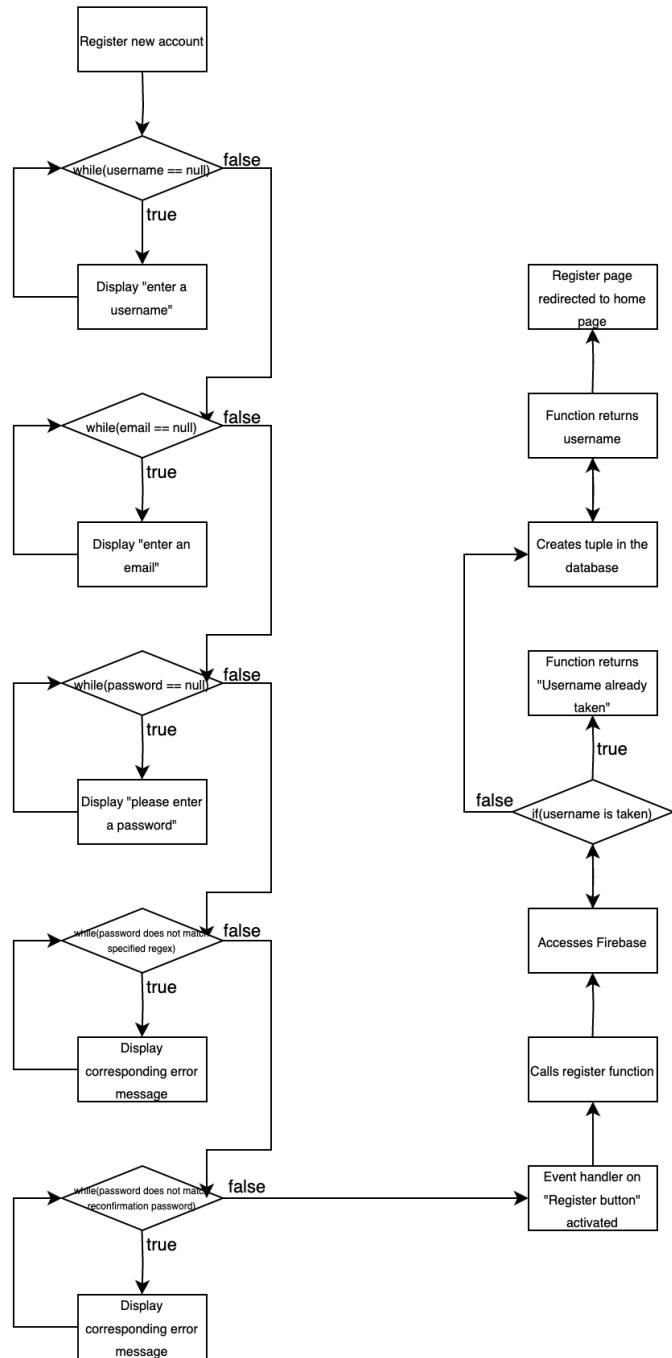
EC5	Change password to a different password that does not contain an uppercase letter but is longer than 8 characters and has a number	Password = "password1"	Error message shows saying that password cannot be changed and password does not change	Error message is displayed saying that 'Password is invalid' and password is not changed	Pass
EC6	Change password to a different password that does not contain a number but is longer than 8 characters and has an uppercase letter	Password = "Passwords"	Error message shows saying that password cannot be changed and password does not change	Error message is displayed saying that 'Password is invalid' and password is not changed	Pass
EC7	Key in different passwords into the New Password and Confirm Password tab	New password = "Pas sword1" and Confirm password = "Password2"	Error message shows saying that the passwords do not match and password does not change	Error is shown on the tab saying that 'Passwords do not match' and password is not changed	Pass
EC8	Change profile picture to a picture in the Photos Gallery and press Save Changes button		Profile picture changes to the selected photo	Profile picture is changed to the selected photo from the Photo Gallery	Pass
EC9	Change profile picture to a picture in the Photos Gallery and does not press Save Changes button		Profile picture reverts back to the old profile photo	Profile picture does not change to the selected photo and it reverts back to the original profile photo	Pass

6.1.7. Report Bugs Functionalities

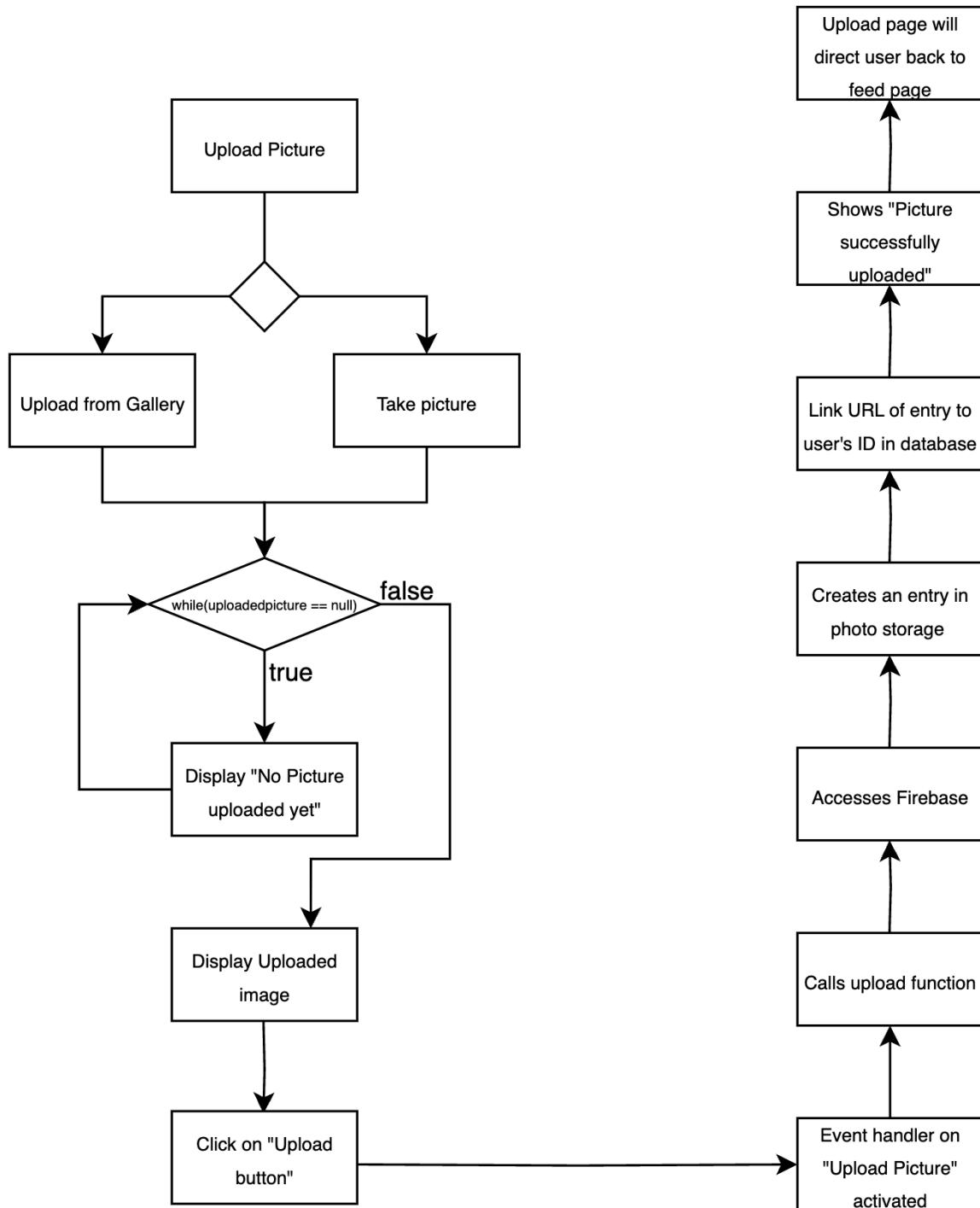
Test ID	Test Case Description	Input Information	Expected Output	Actual Output	Pass/Fail
EC1	File a bug report with a filled title and description	Title = "Hello" Description = "World"	Message showing that the report has been successfully made	Message is displayed saying that 'Your bug report has been submitted successfully'	Pass
EC2	File a bug report with an empty title and filled description	Title = "" Description = "World"	Error message showing that title needs to be filled	Error message is displayed saying 'Please fill in the title'	Pass
EC3	File a bug report with a filled title and empty description	Title = "Hello" Description = ""	Error message showing that description needs to be filled	Error message is displayed saying 'Please fill in the description'	Pass
EC4	Write a description that is over 100 characters long		System prevents further input of characters past 100 characters	Counter exists that shows the current character count and when it hits 100/100, no further inputs can be made into the description textbox	Pass

6.2. White Box Testing

6.2.1. Register New Account



6.2.2. Upload Pictures



7. Appendix A: Glossary

7.1. Data Dictionary

Term	Description
Login details	<p>This includes a unique username created with a minimum of 8 and maximum of 32 alphanumeric characters.</p> <p>This also includes a password with a minimum of 8 and maximum of 32 characters. A combination of at least 1 upper-case letter, 1 lower-case letter, and 1 number is required.</p>
User	The user refers to anyone using the app, regardless if the user possesses an account.
System	In the context of our application, the system refers to both the front-end and back-end portion of our software that is responsible for providing interactive use and manipulating data efficiently.
Registered user	Registered user refers to a user with an account. Registered users can save locations and upload pictures into the application.
Good sunset	A good sunset is characterized by a rich palette of colors, including various shades of orange, red, yellow and purple.
Golden hour quality	A percentage which predicts the quality of sunset happening at a particular day and location. A high percentage means the sunset is likely to be beautiful and a low percentage means that the sunset is likely to be dull.

Location manual	A page in our application where users can search for different locations to check for the predicted quality of sunset and the timing. Registered users have the option to save locations for future use.
Sunset Calendar	A calendar that shows the predicted quality of sunset for each day. User can click in on a date to view more details of the sunset on that particular date.
Follow	A function that lets user A connect to user B, and allow user A to view user B's current and future content easily.
Bugs	An unexpected problem that the user might face, which can include UI issues or other problems interacting with the application that affect the user's experience.

7.2. Sunset Quality Algorithm

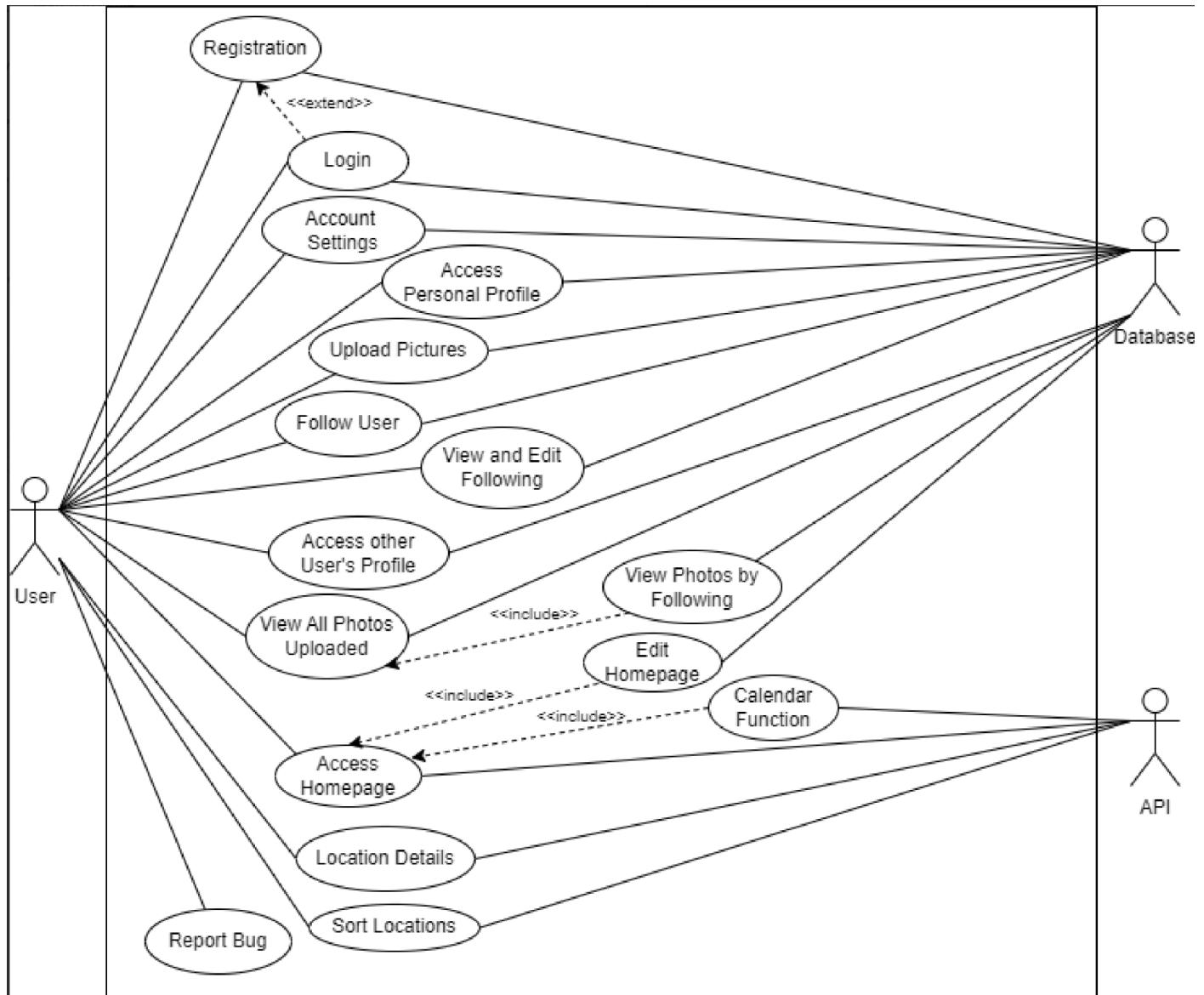
```
cloudCoverQuality = allValues[location][2] * 0.4;  
humidityQuality = allValues[location][1] * 0.3;  
PSI = allValues[location][4].toDouble();  
if (PSI <= 55) {  
    PSIQuality = 80 + ((55 - PSI) / 55 * 20);  
} else {  
    PSIQuality = 20 + ((250 - PSI) / 250 * 80);  
}  
PSIQuality = PSIQuality * 0.3;  
sunsetQuality = cloudCoverQuality + humidityQuality + PSIQuality;  
allValues[location].add(sunsetQuality.toStringAsFixed(1));
```

After researching on how various factors affect sunset, our team came up with an algorithm to predict the quality of sunset. This algorithm predicts sunset quality by considering three main factors: cloud cover, humidity, and the Pollutant Standards Index (PSI).

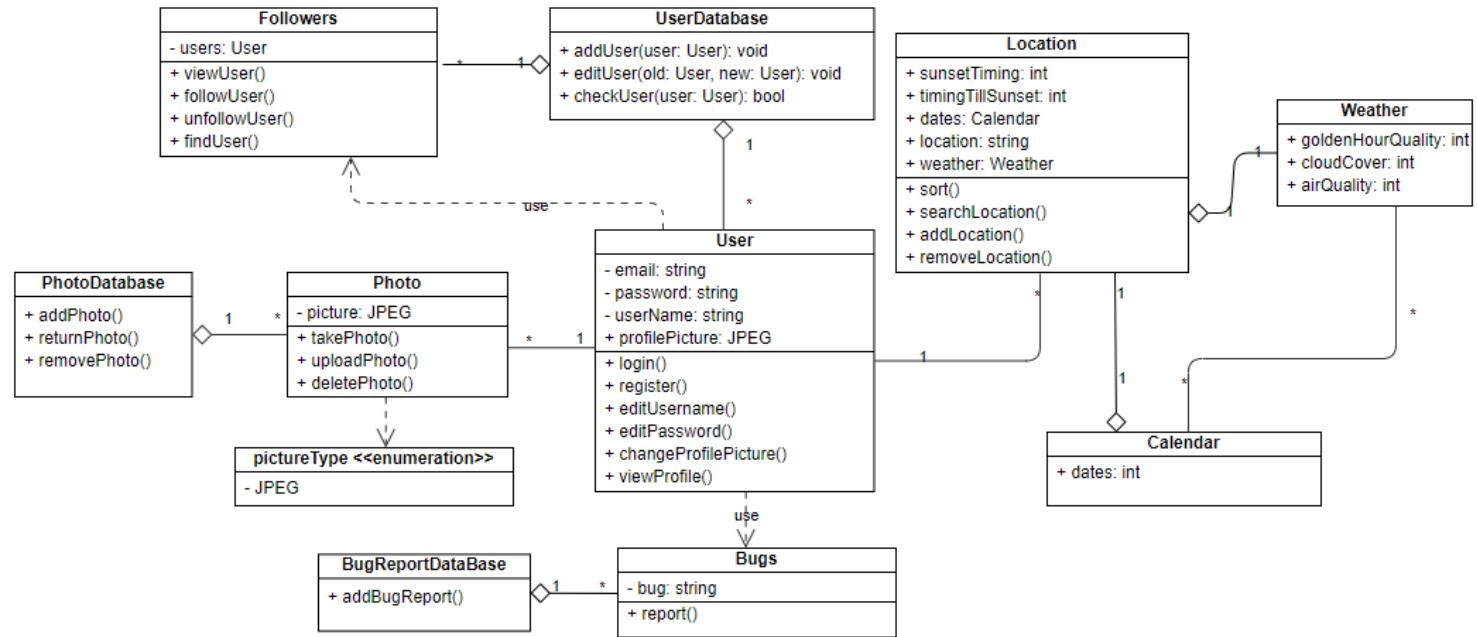
1. **Cloud Cover Quality:** It assesses the extent of cloud cover at the location. The algorithm assigns a weight of 0.4 to this factor.
2. **Humidity Quality:** This factor evaluates the level of humidity in the atmosphere. It is given a weight of 0.3.
3. **PSI Quality:** The PSI, which measures air pollution, is considered. Depending on the PSI value, a quality score is calculated. If the PSI is less than or equal to 55, the quality score is calculated as 80 plus a proportion of the PSI value. Otherwise, it's calculated as 20 plus a proportion of the PSI value. The resulting quality score is then weighted by 0.3.

8. Appendix B: Analysis Models

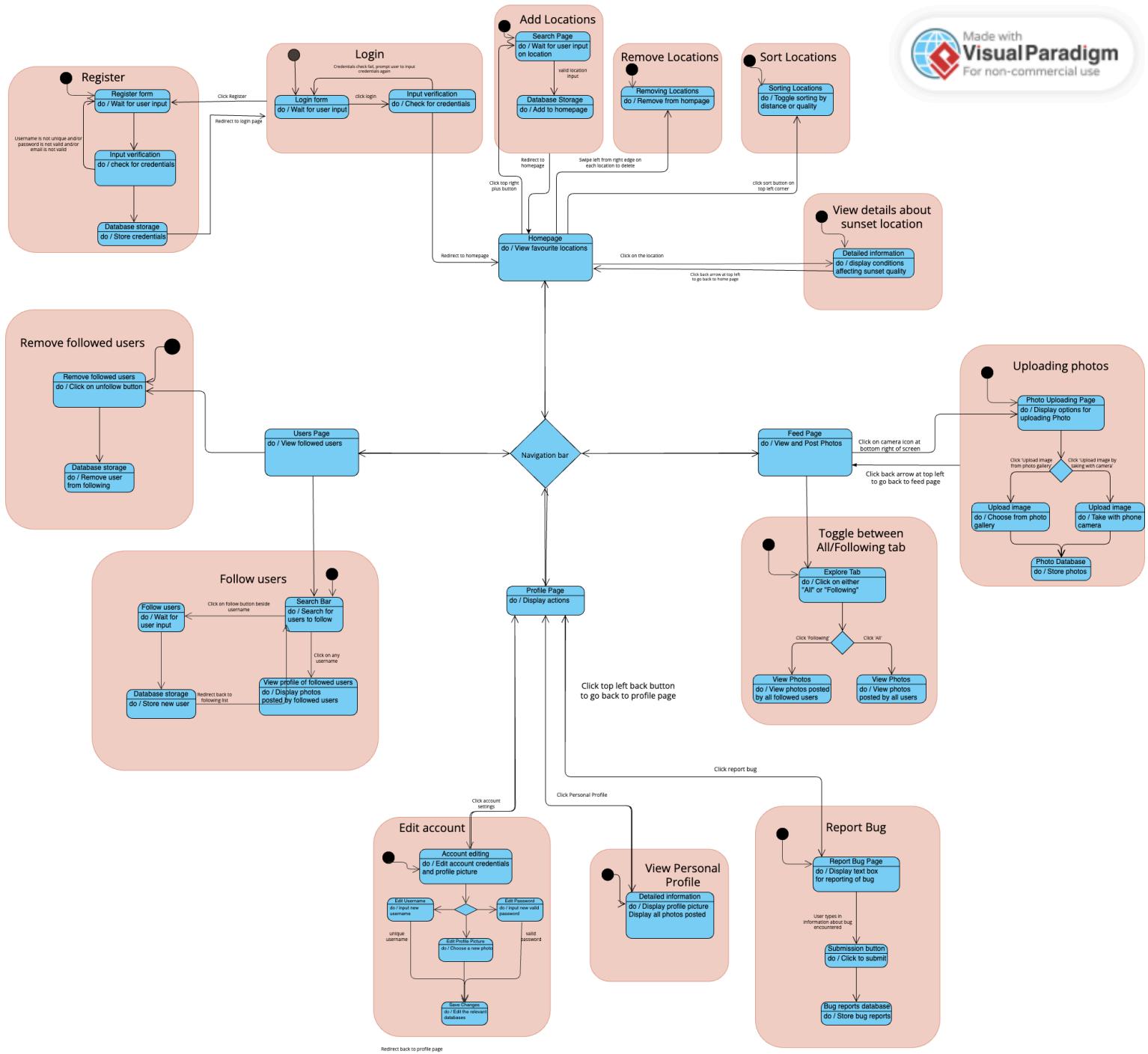
Complete Use Case Model



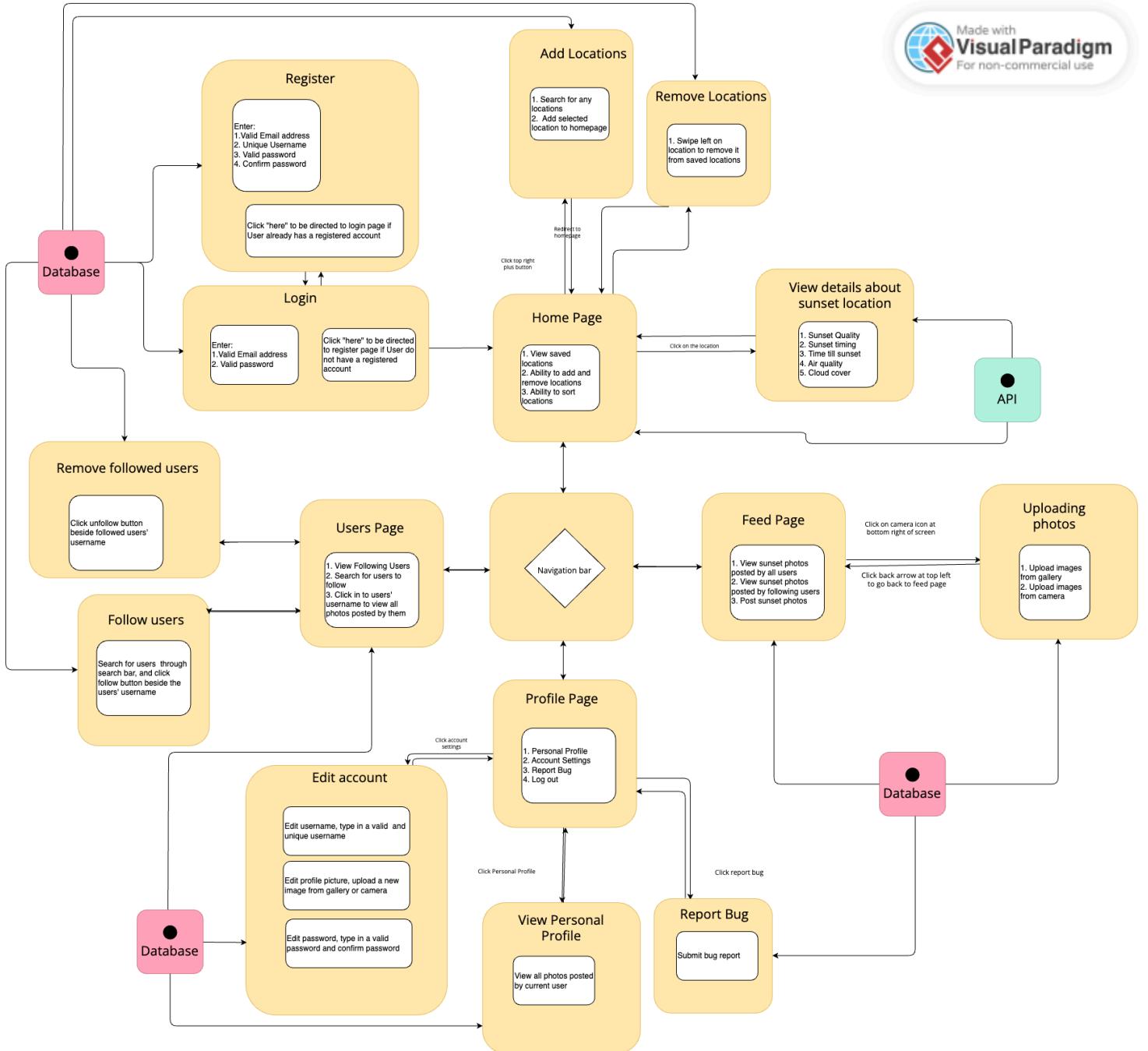
Complete Class Diagram



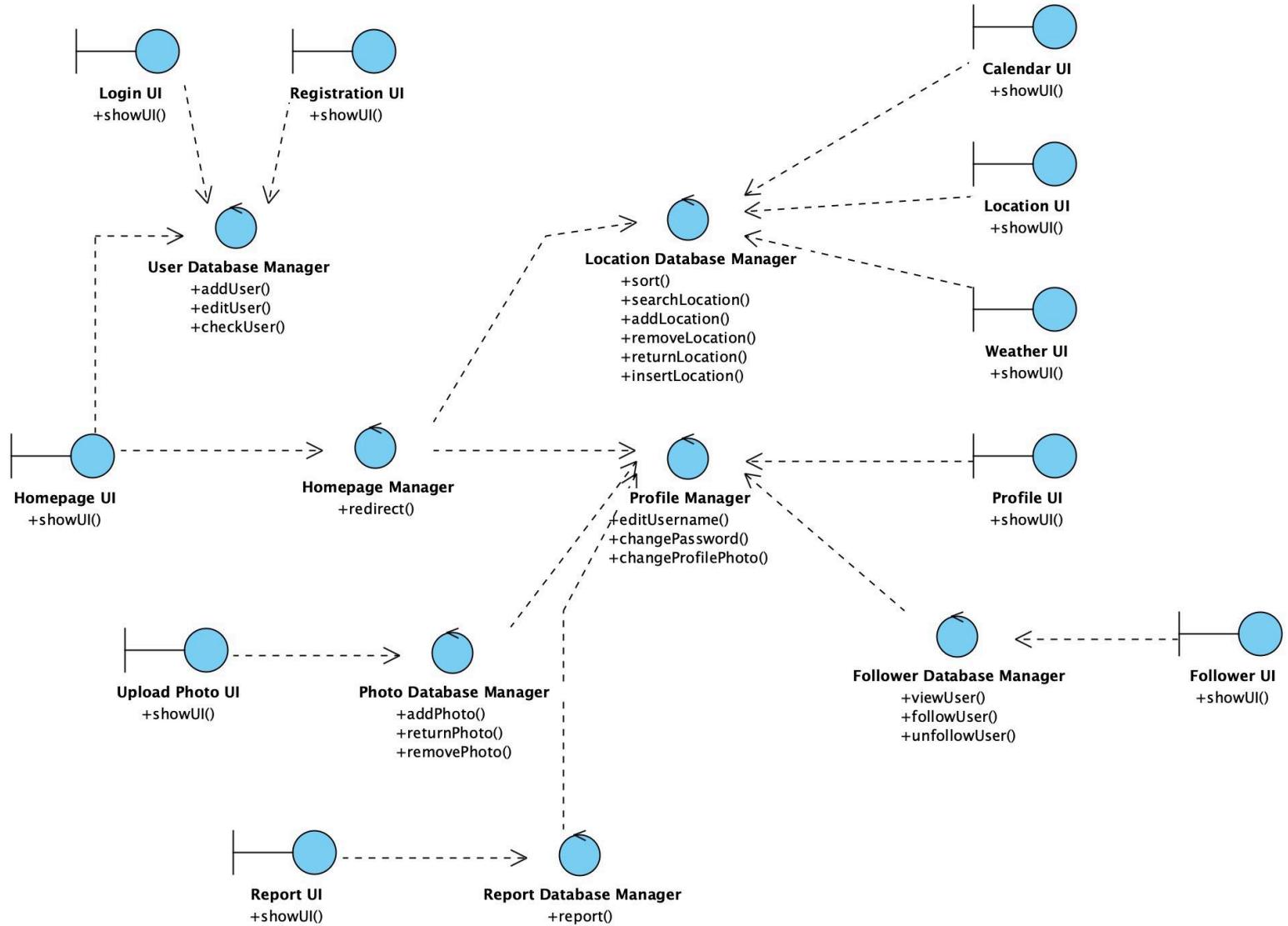
Complete Dialog Map



Complete Application Skeleton

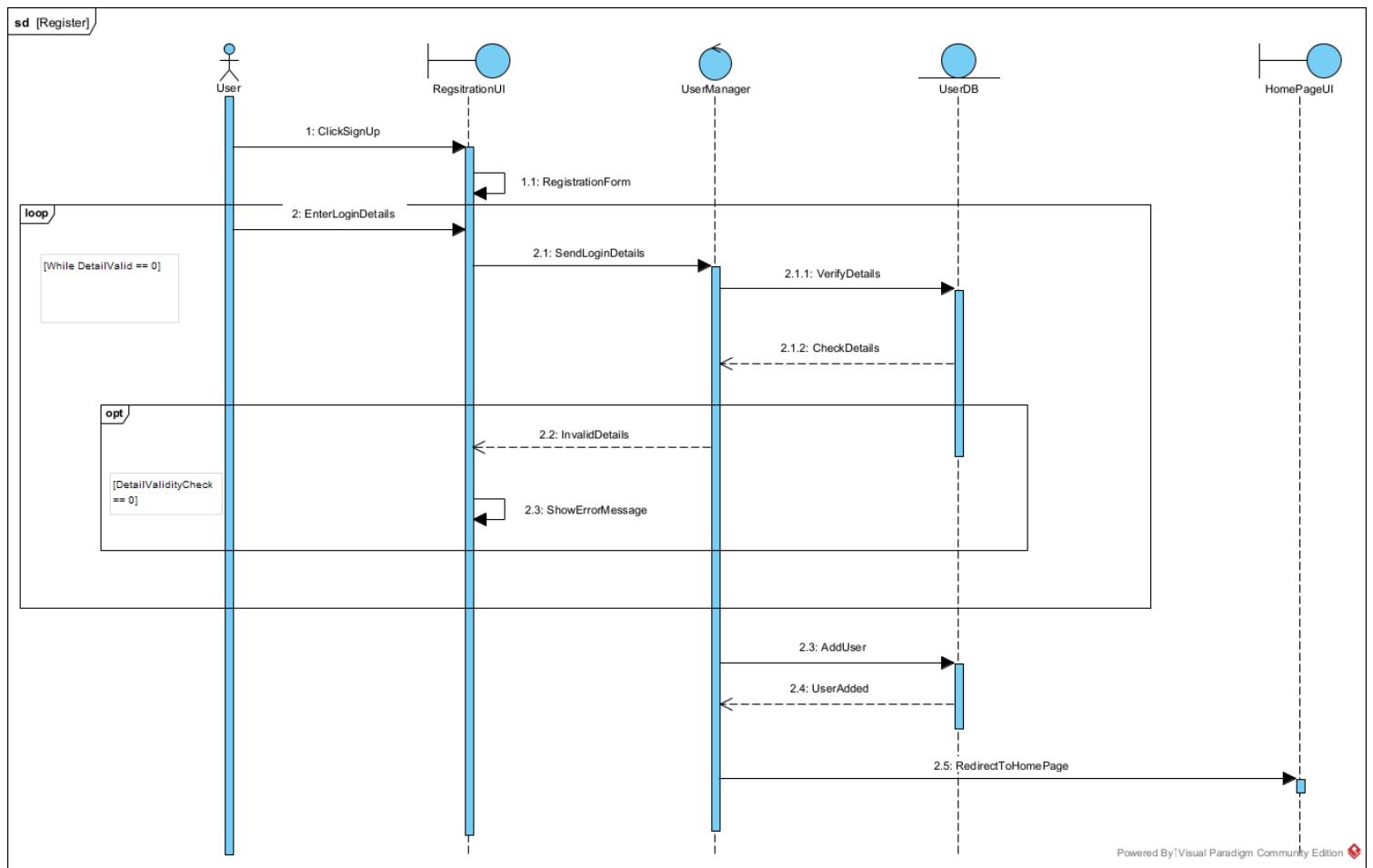


Key Boundary and Control Classes Diagram

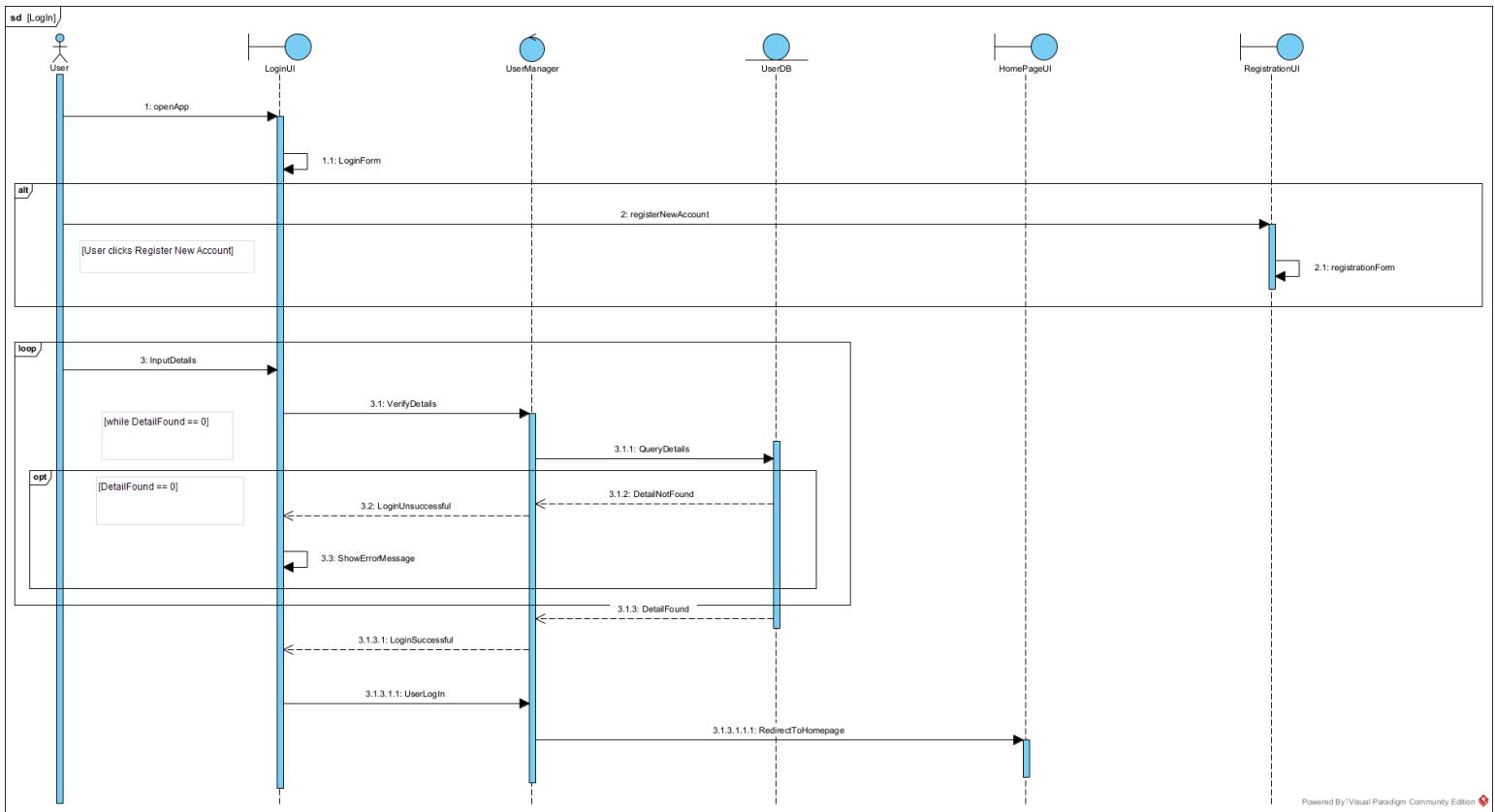


Complete Sequence Diagrams

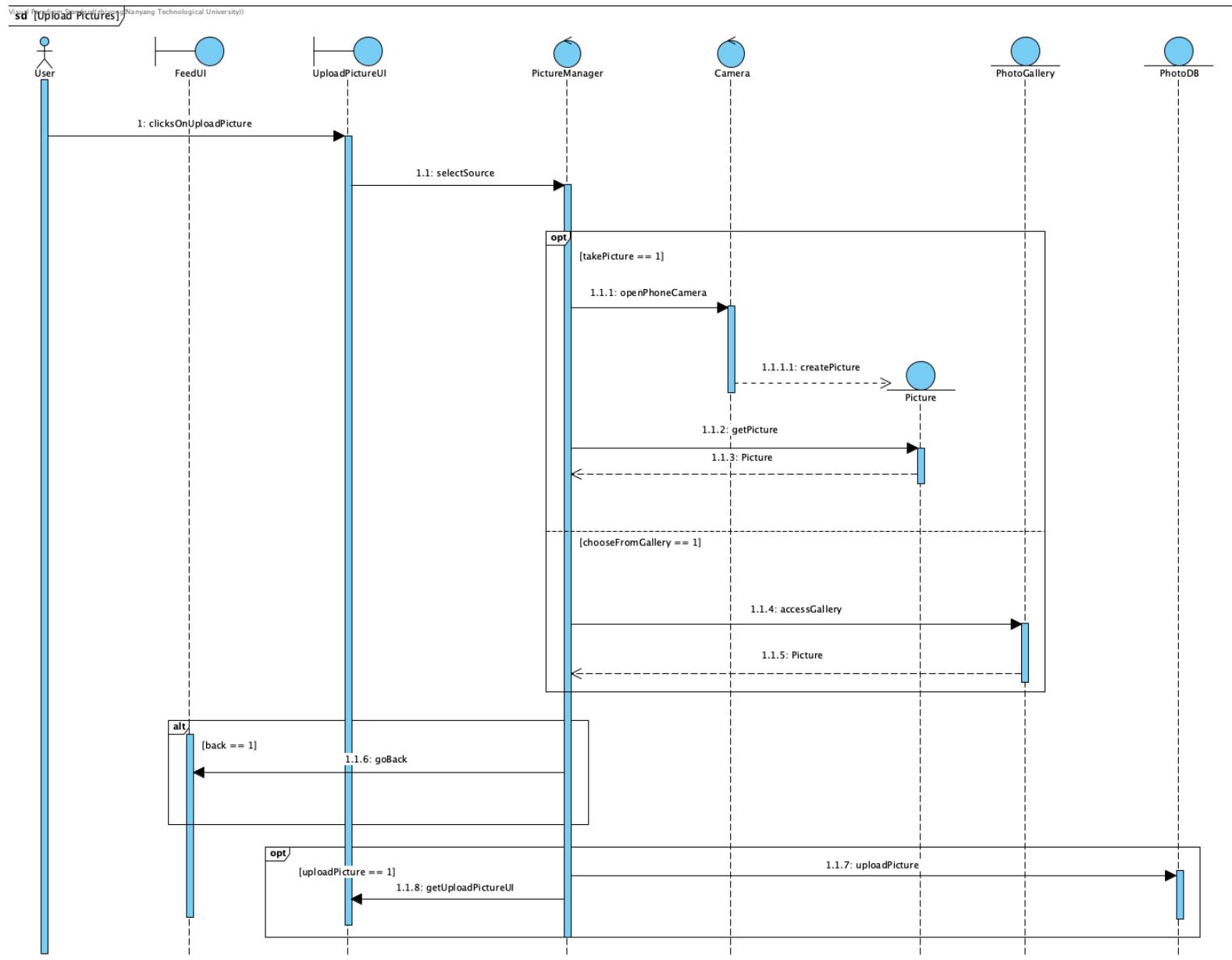
Register



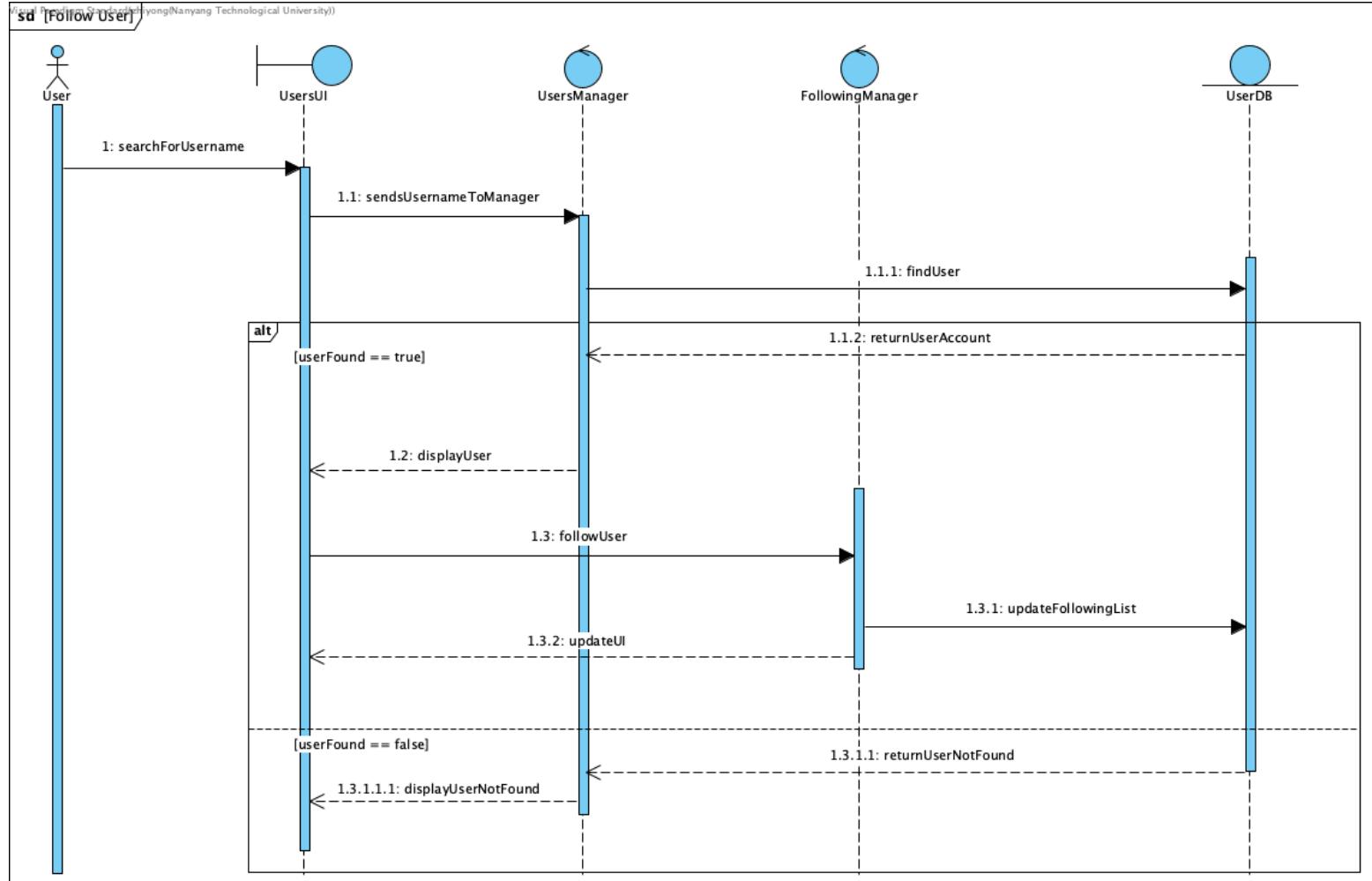
Login

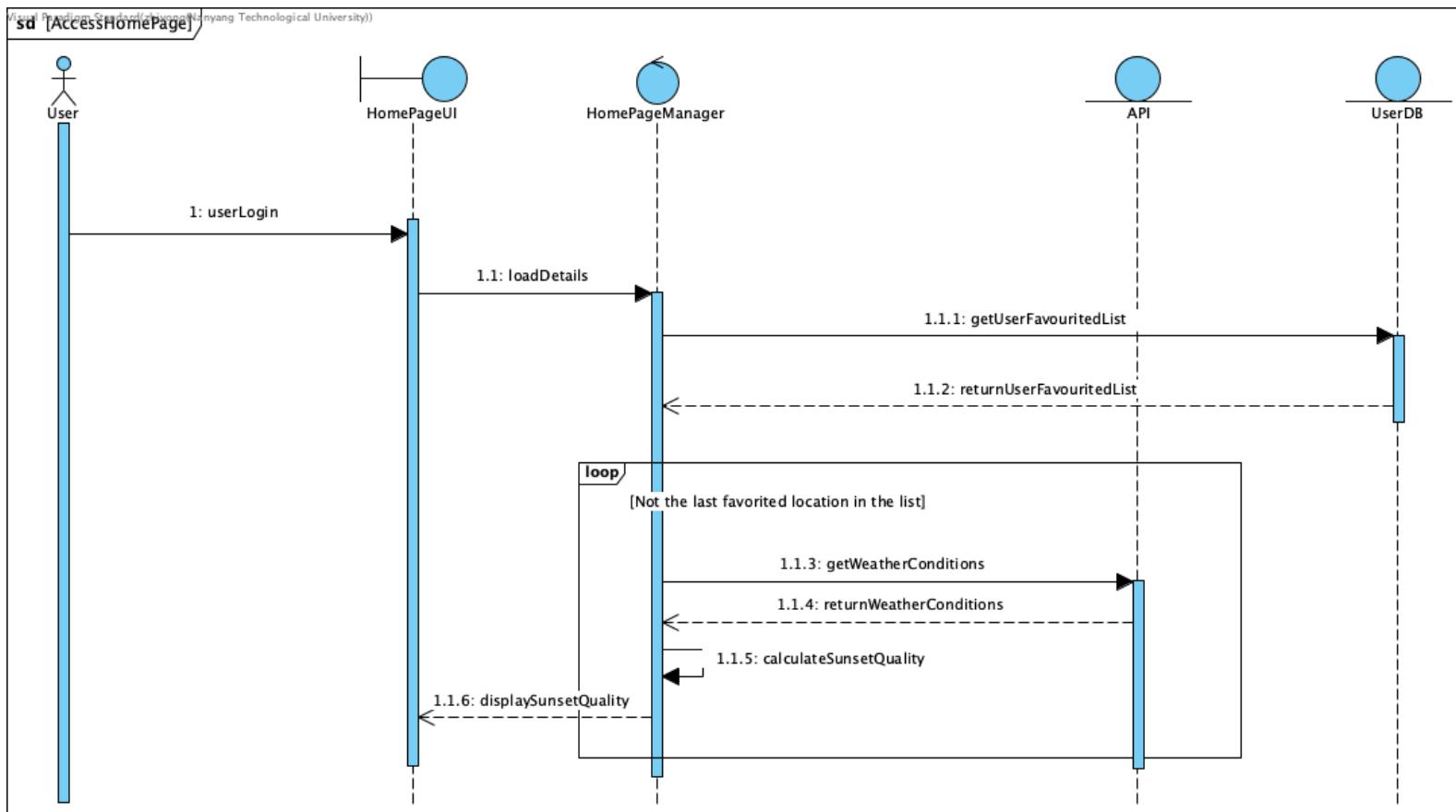


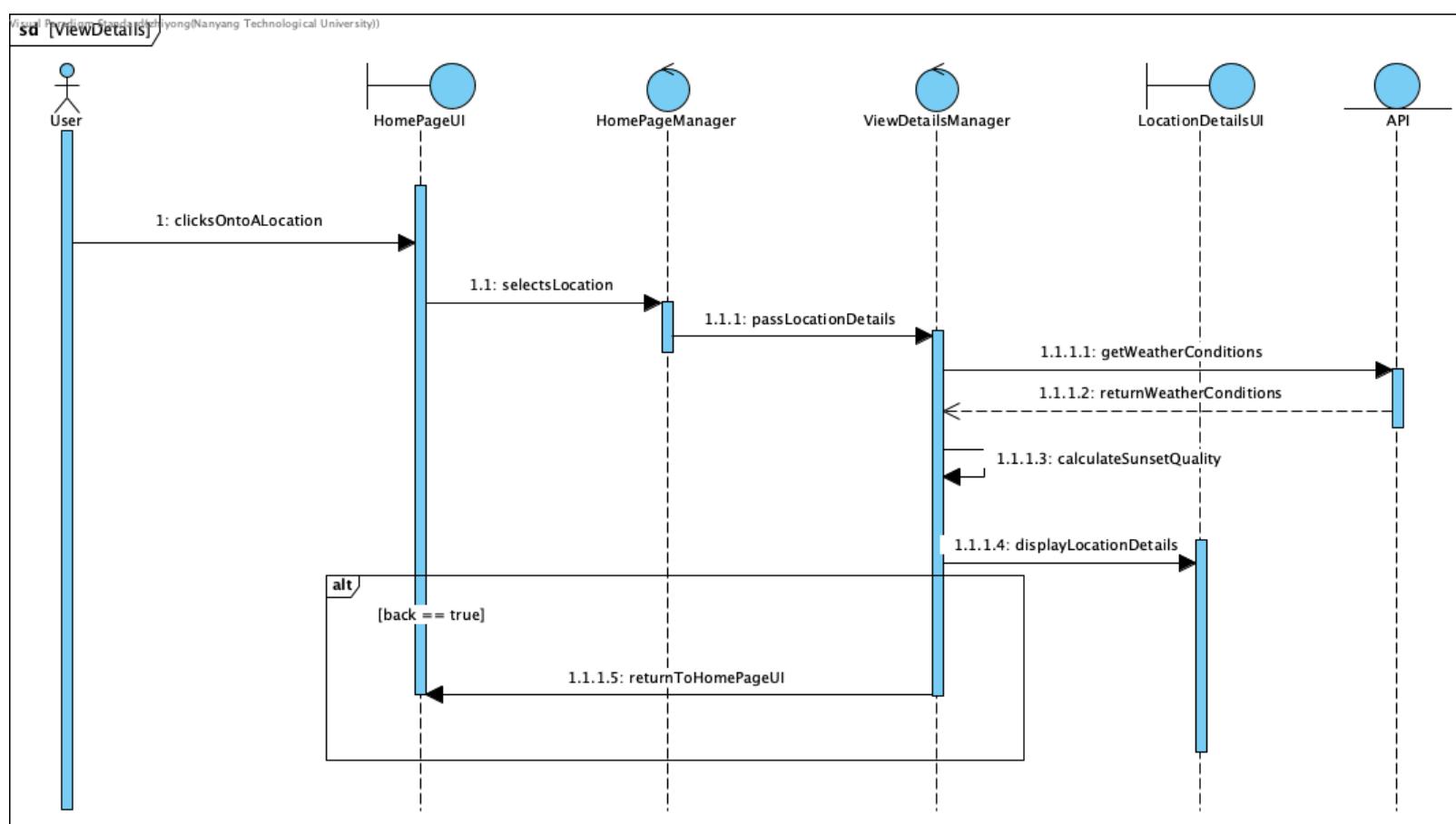
Upload Pictures



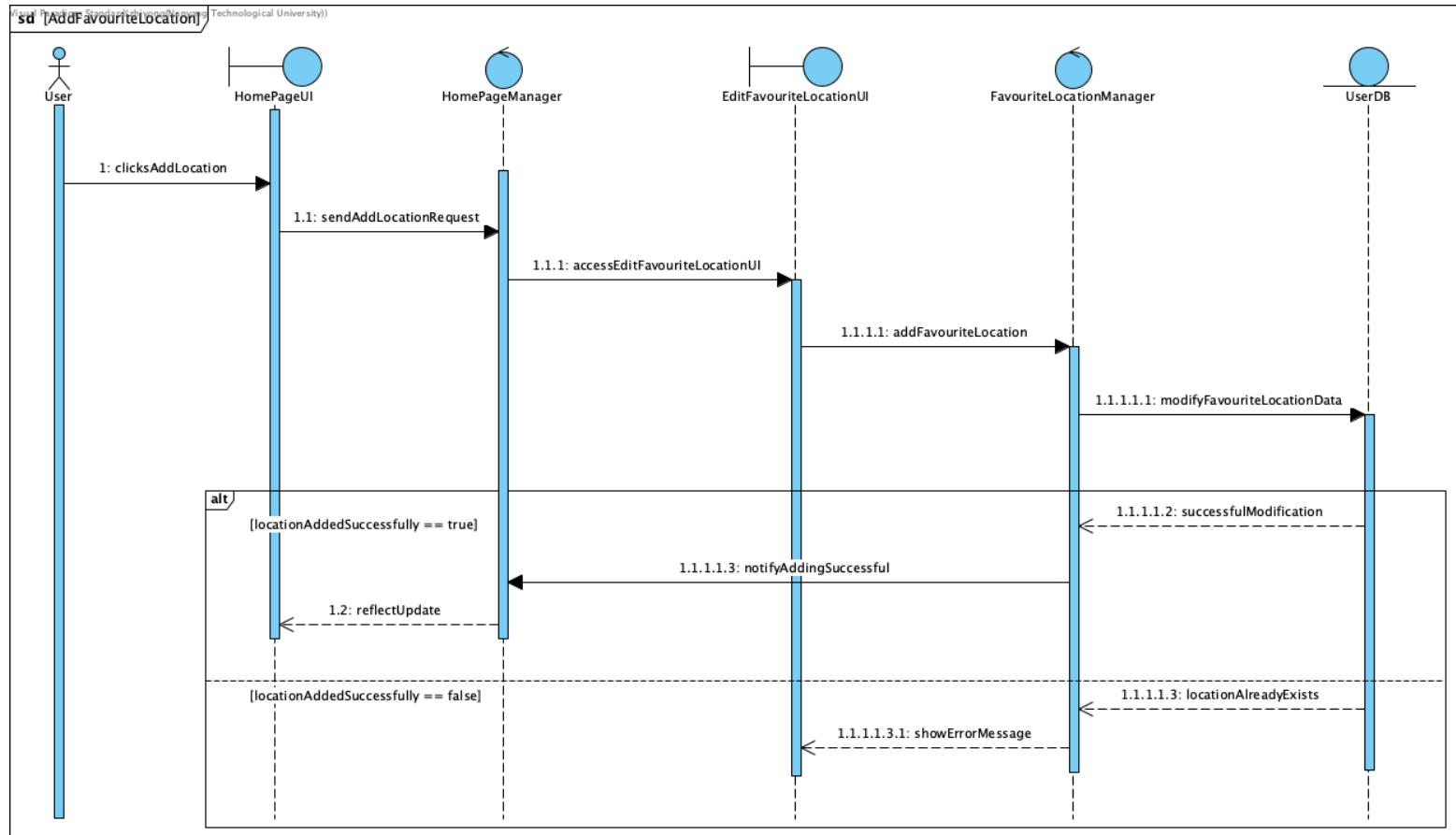
Follow User

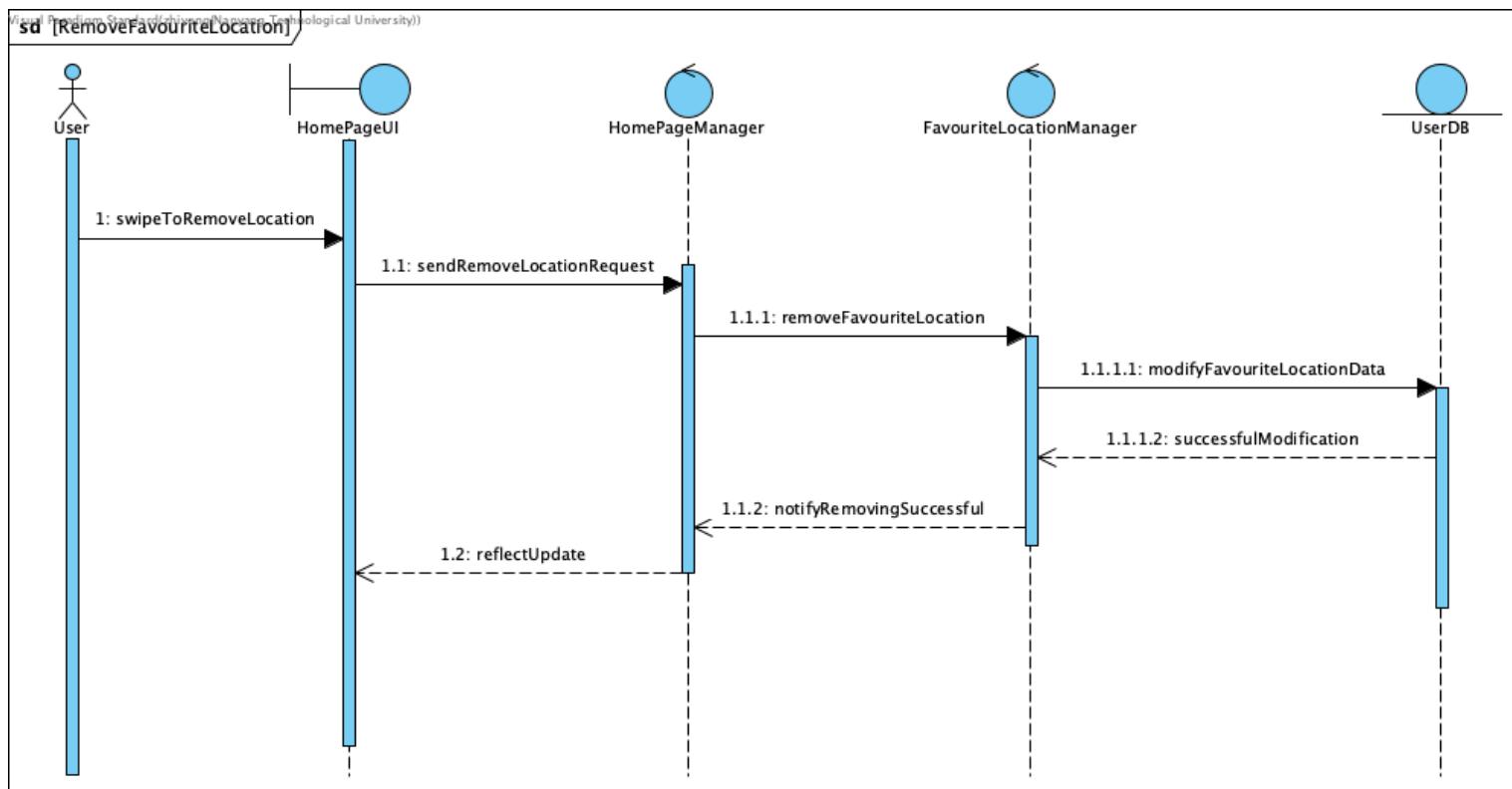


AccessHomePage

ViewDetails

AddFavouriteLocation



RemoveFavouriteLocation

BugReport