Software Requirements Specification

for

LooLah

Version 6.0 approved

Prepared by:

Lim Jun Rong, Ryan

J'sen Ong Jia Xuan

Rachel Lim (Lin Jiahuan)

Deuel Teo Guang Zheng

Chloie Tan Yue Yun

Nanyang Technological University, Clumsy Koalas

<10/09/2023>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Lab 1 Deliverables	10 Sep 2023	Use cases implementation	1.0.0
Lab 2 Deliverables	24 Sep 2023	Class diagram and Sequence diagram	2.0.0
Lab 3 Deliverables	29 Oct 2023	Dialog Map, System Architecture and Application Skeleton	3.0.0
Lab 4 Deliverables	12 Nov 2023	Application and Test Cases	4.0.0
Lab 5 Deliverables	18 Nov 2023	Live Demo and Deliverables	5.0.0
Final	19 Nov 2023	Final Edits	6.0.0

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a full in-depth description of LooLah– a mobile platform toilet option in Singapore. This SRS describes a single subsystem which defines the system boundaries, interface and communications used with the external APIs and how it interacts with our application. It also includes the scope of the project, an overall description of the project, interface requirements, functional requirements, and non-functional requirements.

1.2 Intended Audience and Reading Suggestions

This SRS document is intended for developers, project managers, investors, testers and general users. Developers and project managers are recommended to read the entirety of this document. Investors, testers and general users are recommended to read the overall description and system features pertaining to the app in order to understand the features of this app.

1.3 Product Scope

LooLah is a community-based app that helps navigate users to nearby toilets. In addition to providing directions, the app features a toilet rating system that allows users to rate and review toilets. This ensures that users have access to more information about the toilet they plan to visit, allowing them to make a more informed choice and find a toilet that better suits their needs. Through LooLah, users can attain a satisfactory public restroom experience.

1.4 References

Integrating Firebase with android 2022.3.1
https://firebase.google.com/docs/android/setup

Android Studio Documentation
https://developer.android.com/docs

Jakob Nielsen's 10 general principles for interaction design
https://www.nngroup.com/articles/ten-usability-heuristics/

2. Overall Description

2.1 Product Perspective

"LooLah'' is an innovative toilet-finding app that addresses the common problem of inadequate restroom information. Our application integrates with Google Maps and data.gov.sg datasets to provide up-to-date insights into nearby toilets in Singapore. Our mission is to enhance users' daily lives by offering a satisfactory restroom experience and fostering community engagement through user ratings and reviews. The app prioritises good software engineering principles, implementing an intuitive UI, personalization features, and a robust backend using Firebase and the MVVM design pattern. Key functionalities include the Home Page, Map Page, Toilet Details Page, Favourites Page, and Add Reviews Page, catering to user convenience and engagement.

The major components of our application are defined in the System Architecture in Figure 2.1.1. It is composed of a Model-View-ViewModel (MVVM) architecture. The MVVM separates the data (Model), the presentation (ViewModel) and the UI (View) into distinct modules. This approach makes the code more modular and easier to maintain, resulting in cleaner and more readable code, and facilitating collaboration.

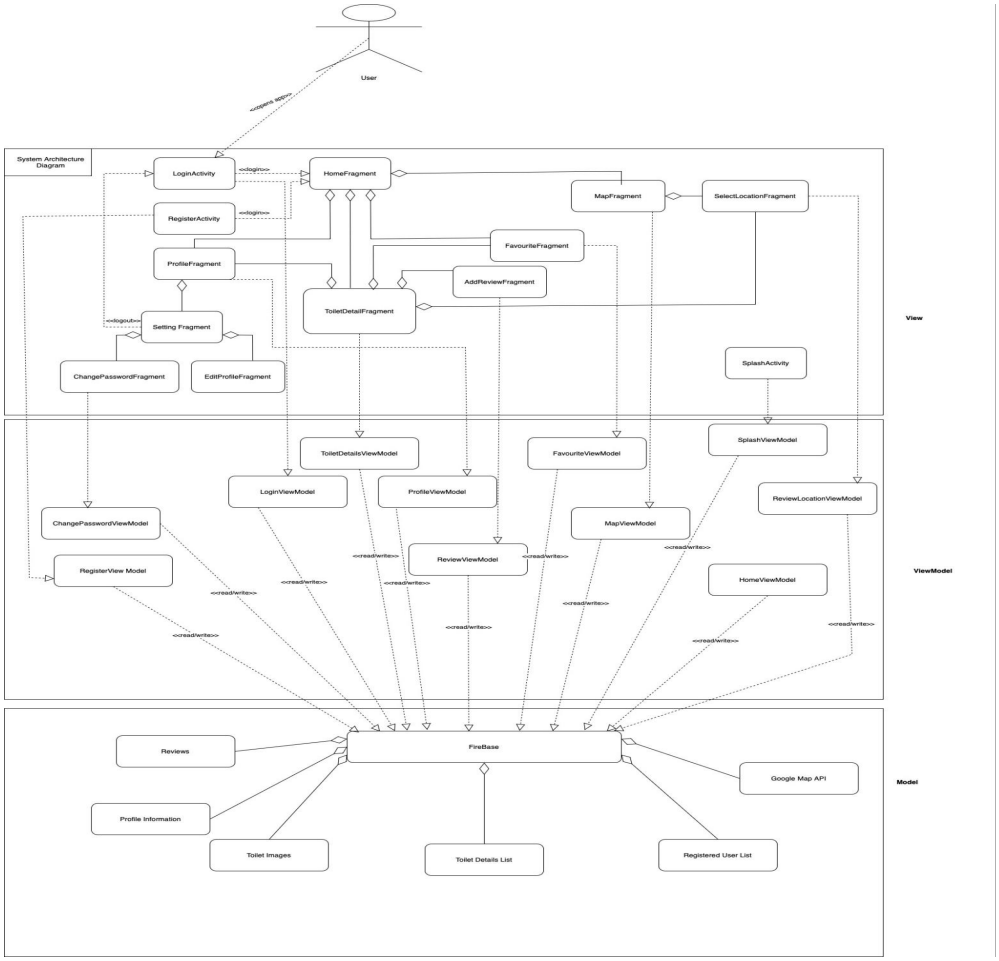


Figure 2.1.1 System Architecture

2.2 Product Functions

2.2.1 Functional Requirements

1. Users shall be able to register an account and sign in to the application:

1.1. Users must create an account by entering their email/phone number, username and password.
1.2. Users shall log into their accounts by entering their respective emails and passwords.
1.3. Users can change their password with authentication
1.4. If the user wishes to change their password, the user will be able to do so through the application settings.
1.5. Users will be able to change their username if the desired username is not already taken by another user.

2. Users shall be able to personalise their account profiles:

2.1. Users shall be able to customise their profiles
2.2. Profile picture
2.3. Username

3. Users shall be able to search for nearby toilets:

3.1. Users shall be able to search for nearby toilets based on their location.
3.2. The application will provide navigation guidance to the selected toilet.
3.3. The application will display the toilet name, address, pictures, descriptions, crowd level, accessibility and ratings (1 - 5 stars).

4. Users shall be able to rate and review toilets:

4.1. Users shall be able to rate toilets on a scale (1 - 5 stars) based on their experience.
4.2. Users shall be able to rate their overall user satisfaction based on the cleanliness/design/maintenance of the toilet.
4.3. Users shall be able to post detailed reviews and provide feedback on their overall experience.
4.4. Users can view and like other users' reviews.

5. Users shall be able to edit their reviews:

5.1. Users shall have the option to edit their past reviews.
5.2. Users shall have the option to delete their past reviews.

6. Users shall be able to favourite toilets:

6.1. Users shall have the option to add a toilet to their favourites list.
6.2. Users shall have the option to remove a toilet from their favourites list.

7. Users shall be able to add new toilets:

7.1. Users shall have the option to submit new toilets to the application.
7.2. Users shall have the option to upload images of the toilet to the application.

8. Users shall be able to view their profile:

8.1. Users shall be able to view a history of their past reviews and ratings.

9. Users shall be able to filter the type of toilets:

9.1. Users shall be able to filter the type of toilets

9.1.1. Type of toilets (Gender, Handicap, Child-friendly)

9.1.2. District (Central, North East, North West, South East, South West)

9.1.3. Distance from user's current location (within 500m)

9.1.4. Rating (1-5 stars)

10. Users shall be able to get directions to their selected toilet:

10.1. The application shall navigate users by providing directions to the selected toilet.

2.2.2 Non-functional Requirements

1. User experience:

1.1. Application must have an intuitive and user-friendly interface.

1.2. Navigation to different sections must be seamless and intuitive.

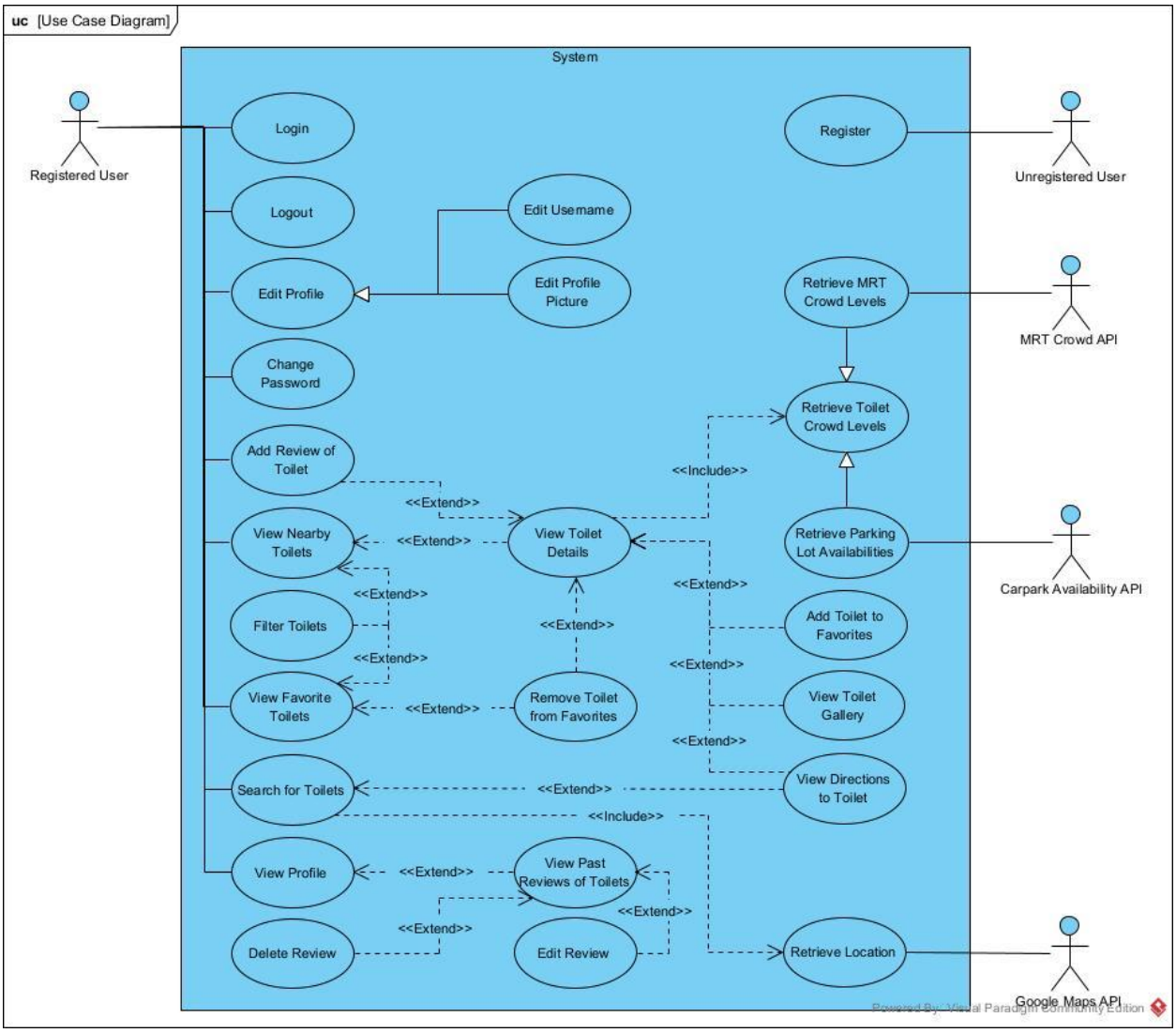
2. Performance:

2.1. Application will be able to handle x amount of concurrent users even during peak usage. (e.g. during breakfast/lunch/dinner time)

2.2. New toilets/Toilet ratings will be updated in real-time.

2.3. Requests will be processed within 3 seconds.

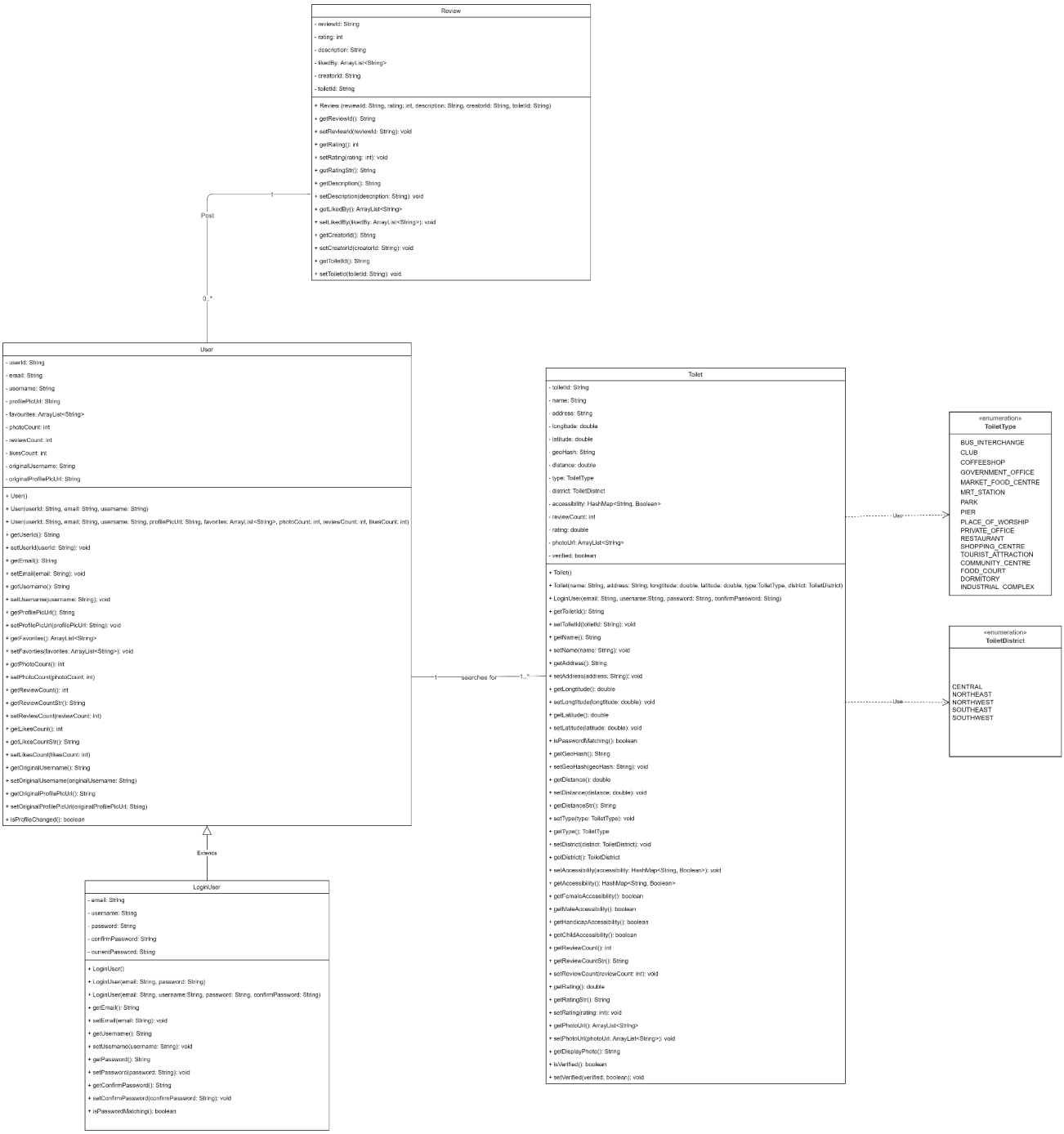
2.2.3 Use Case Diagram



Major functionalities:

1. Add review of toilet
2. View nearby toilets
3. View toilet details
4. Search for toilets
5. View direction to toilet
6. Add toilet to favourites

2.2.4 Class Diagram



2.3 User Classes and Characteristics

2.3.1 Regular Users

Frequent users who rely on “LooLah” for daily navigation to nearby toilets. They primarily use the app to find nearby toilets and may occasionally contribute to the rating system. They may not engage extensively with the rating and review features.

2.3.2 Contributing Users

Users who add to the “LooLah” database by contributing to toilet reviews at new locations. They are individuals who rely on the app regularly to consistently find well-reviewed restrooms, possibly due to their specific needs or preferences.

2.3.3 Moderators

Moderators regularly use “LooLah” to manage user-generated content and ensure its accuracy and appropriateness. They are responsible for overseeing and moderating user-generated reviews and ratings.

2.4 Operating Environment

Our application is developed entirely in Java using Android Studio Giraffe | 2022.3.1 Patch 4 and operated on the Android Operating System for Android 9 and above. Our application dataset is hosted on Firebase, utilising the Firestore Database, Storage and Authentication the platform provides.

Our application integrates Google Maps and Geocoding API to enhance our map features. Additionally, we leverage real-time data from LTA Carpark Availability and MRT Platform Crowd Density APIs to provide users with insights into the toilet crowd levels, addressing the absence of readily available crowd-level APIs.

Google Maps must be installed in the Android Operating System to utilise the navigation functionality of our application.

2.5 Design and Implementation Constraints

1. Our application is designed to only run on devices powered by the Android operating system. Developers will need to concentrate on utilising Android-specific tools, SDKs (Software Development Kits), and APIs provided by Google for further app development.

2. Our application architecture follows a Model-View-ViewModel (MVVM) architecture. Maintaining the application would require an understanding and familiarity of this architecture.

3. There is a constraint on the number of users we can have running our application concurrently. This is because we are using the basic plan of Geocoding API and there is a quota limit on daily query submissions. In the event of a substantial increase in our user base, this limitation could potentially lead to operational challenges.

2.6 User Documentation

LooLah capitalises on the familiarity of widely used social media platforms by adopting their established user interfaces. The inclusion of a navigation bar at the bottom of the screen with familiar icons depicting their respective functions. For example, an icon resembling a map links to our application's map function. This design approach eliminates the need for a separate tutorial to use our application.

2.7 Assumptions and Dependencies

The following assumptions were made to ensure a working system:

1. The Firebase server will not be down/under maintenance when our application is being used.
2. Google Maps will not be down/under maintenance when our application is being used.
3. The APIs used by our application will not be modified in which our application will not have access to the data provided by the APIs.
4. Users are always connected to the internet.
5. Users will allow location permission and will always have location turned on.
6. Users have smartphones that have compatible Android operating systems.

3. External Interface Requirements

3.1 User Interfaces

We harnessed Jakob Nielsen's 10 general principles known as "**Nielsen Heuristics**" for interaction design. These heuristics serve as guidelines for creating user-friendly and effective user interfaces. We kept his postulations in mind when designing LooLah's interface.

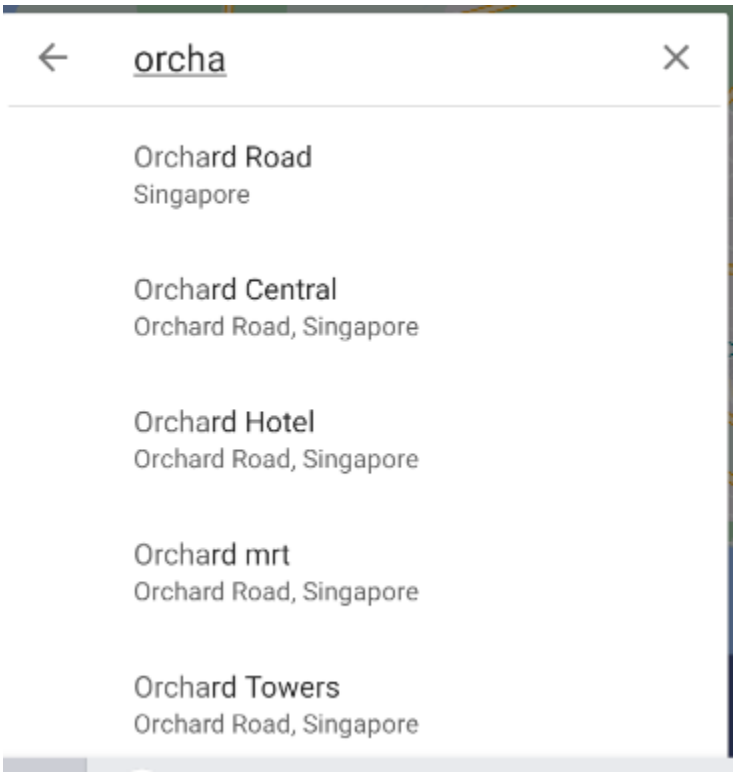
3.1.1 Navigation Bar

We incorporated recognizable symbols that align with real world conventions within the navigation bar, to facilitate user understanding and navigation, adhering to the principle of "Matching between the system and the real world".

3.1.2 Buttons

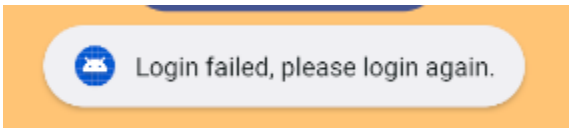
Buttons are deliberately placed and implemented with universally recognisable symbols for easy recognition of function, minimising the need for users to recall specific actions, this is in line with the principle of "recognition rather than recall".

3.1.3 SearchView



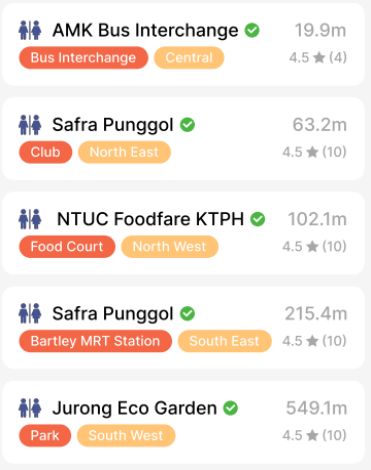
By implementing a searchview functionality, users can efficiently navigate and retrieve information aided by auto-suggestions. This reduces the likelihood of errors, abiding to the principle of “Error prevention” .

3.1.4 Toast messages



Clear and concise error messages are provided to the user in the form of toast messages, assisting users in understanding issues and providing steps for troubleshooting. This aligns with the principle of “Helping users to recognise, diagnose and recover from errors”.

3.1.5 RecyclerView



By implementing Recyclerview into our application, users are able to scroll freely and explore the list of toilets at their own pace, this is in line with his third principle “User Control and Freedom”.

3.1.6 Spinner

The spinners allow users to customise their searches using specific filters. Users will be able to search based on the different types of toilets, district area, how far the toilet is from them and the rating of the toilet. This is in line with his third principle “User Control and Freedom”.

3.2 Hardware Interfaces

LooLah requires Android 9 or later devices.

3.3 Software Interfaces

Dataset used in the application:

1. toilet.org.sg/happytoilets dataset - To retrieve toilet information and details Dataset.

APIs used in the application:

1. Google Map API - To integrate Map functionalities in our app.
2. Firebase Authentication - To store, retrieve and update toilet dataset, reviews, and user information.
3. Firebase Real-Time Database - To store, retrieve and update toilet dataset, reviews, and user information.
4. Firebase Storage - To store, retrieve and update toilet dataset, reviews, and user information.

3.4 Communications Interfaces

- Communication with the APIs uses HTTPS protocol.
- Android Studio provides libraries with HTTPS native support

4. System Features

4.1 Use Case Descriptions

Use Case ID:	1		
Use Case Name:	Register		
Created By:	J'sen	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Unregistered User (Initiating Actor)
Description:	Unregistered user registering for an account
Pre-conditions:	1. User does not have a registered account, based on their email 2. User is at the login page
Post-conditions:	1. Account is successfully created
Priority:	High
Frequency of Use:	1 - 3 times per lifetime
Flow of Events:	1. Unregistered User selects "Sign Up" on the login page 2. Application redirects user to register page 3. User inputs the required information a. Email b. Username c. Password d. Confirm Password 4. User clicks the "Register" button 5. System validates the entered information a. Email: Valid and not linked to existing accounts b. Username: Unique and meets any specified requirements (e.g., length, characters allowed) c. Password: Meets security criteria (e.g., length, complexity) d. Confirm Password: Matches password entered 6. If the information is successfully validated, the system creates an account with the entered information 7. Application redirects the user to the home page
Alternative Flows:	1-AF-5a: Invalid email entered 1. Application displays an input error message "Invalid email" 2. System returns to step 3

	1-AF-5a: Existing email entered 1. Application displays input error message “Registration failed” 2. System returns to step 3 1-AF-5b: Invalid username entered 1. Application displays input error message “Invalid username” 2. System returns to step 3 1-AF-5c: Password Strength is too low 1. Application displays input error “Invalid password”. 2. System returns to step 3 1-AF-5c: Entered passwords do not match 1. Application displays an input error message “Password does not match.” 2. System returns to step 3
Exceptions:	1-EX-1: User chooses to abandon the registration process at any point and navigates away from the registration page
Includes:	-
Special Requirements:	-
Assumptions:	1. User has an email that is not linked with a registered account
Notes and Issues:	-

Use Case ID:	2		
Use Case Name:	Login		
Created By:	J'sen	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating actor)
Description:	Registered user logging into the application
Pre-conditions:	1. User has an existing account 2. User has a username/email and password associated to an existing account 3. User is at the login page
Post-conditions:	1. User logins successfully
Priority:	High
Frequency of Use:	0 - 3 times per 3 months
Flow of Events:	1. User inputs their email and password 2. User clicks the "Login" button 3. System verifies the email and password with existing accounts in the database 4. If the account is successfully verified, the application redirects the user to the home page
Alternative Flow:	2-AF-3: Invalid username and password 1. The application displays the error message "Login failed" 2. Application returns to step 1
Exceptions:	2-EX-1: User forgets their username/password
Includes:	-
Special Requirements:	-
Assumptions:	1. User knows their username/email and the corresponding password
Notes and Issues:	-

Use Case ID:	3		
Use Case Name:	Logout		
Created By:	J'sen	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user logging out of the application
Pre-conditions:	1. User is logged into the application 2. User is at the account settings page
Post-conditions:	1. User successfully logged out 2. Application directs the user to the login page
Priority:	Low
Frequency of Use:	0 - 3 times per 3 months
Flow of Events:	1. Users clicks the "Sign Out" button in account settings page 2. Application displays a prompt "Are you sure you want to logout?" 3. User clicks the "Confirm" button 4. Application redirects the user to the login page
Alternative Flows:	-
Exceptions:	3-EX-3: User decides not to logout when prompted by the application 1. User clicks the "Cancel" button 2. Application prompt is closed and user is back at the account settings page
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	4		
Use Case Name:	Edit Profile		
Created By:	Chloie	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user can edit username or profile picture
Pre-conditions:	1. User is logged into the application 2. User is at the profile page
Post-conditions:	1. User's profile is updated successfully
Priority:	Low
Frequency of Use:	3 - 5 times per 3 months
Flow of Events:	1. User clicks the "Settings" button 2. User clicks the "Edit Profile" button 3. Edit Username (Optional): User enters a new username 4. Edit Profile Picture (Optional): User clicks on the "Upload" button and selects an image from their device 5. User clicks on the "Save" button 6. System validates the entered information: a. Username: Unique and meets specified requirements (e.g., length, characters allowed) b. Profile Picture: Within size and format constraints 7. If the information is successfully validated, the system updates the user's profile with the new information 8. Application redirects user back to profile page with success message "Successfully updated." and updated information
Alternative Flows:	4-AF-6a: Invalid username entered 3. Application displays input error message "Invalid username" 4. System returns to step 3 4-AF-6b: Uploaded picture does not fit constraints 1. File size: Application displays input error "Image is too large. Maximum image size is 2Mb." 2. File extension: Application displays input error "Image could not be uploaded. Only files with the following extensions are accepted: jpg, png, jpeg." 3. System returns to step 4

Exceptions:	4-EX-1: User chooses to abandon the edit profile process at any point and navigates away from the edit profile page 1. User clicks the “Back Arrow” button 2. Application redirects user to the account setting page 4-EX-2: User made no changes to the username and profile picture 1. User clicks on the “Save” button 2. Application redirects user back to profile page with information message “No changes made.”
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	5		
Use Case Name:	Change Password		
Created By:	Chloie	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user can change password
Pre-conditions:	1. User is logged into the application 2. User is at the settings page
Post-conditions:	1. User's password is updated successfully
Priority:	Low
Frequency of Use:	3 - 5 times per 3 months
Flow of Events:	1. User clicks the "Change Password" button 2. User enters their current password and then the new password twice for confirmation 3. System validates the entered information: a. Current Password: Matches user's password in database b. New Password: Meets security criteria (e.g., length, complexity) c. Confirm Password: Matches entered new password 4. If the information is successfully validated, the system updates the user's profile with the new information: a. If the user changed their password, it updates the password with the new one. 5. Application redirects user back to profile page with success message "Password successfully updated."
Alternative Flows:	5-AF-3a: Invalid current password 1. Application displays an input error message "Current password incorrect." 2. System returns to step 2 5-AF-3b: New password strength is too low 1. Application displays input error "Password is too weak.". The application will then provide guidance on improving password strength (e.g., length, complexity, or combination of upper and lower - case letters)

	2. System returns to step 2 5-AF-3c: Entered passwords do not match 1. Application displays an input error message “Passwords do not match.” 2. System returns to step 2 5-AF-3: Empty input fields for username or new password 1. Username: Application displays an input error message “Username is required. Please enter a valid username.” 2. New Password: Application displays an input error message “Please enter a new password.” 3. System returns to step 2
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	6		
Use Case Name:	View Profile		
Created By:	Chloie	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user can view their profile
Pre-conditions:	1. User is logged into the application
Post-conditions:	1. System directs the user to their profile page
Priority:	Low
Frequency of Use:	0 - 3 times every month
Flow of Events:	1. User selects “Account” in the application’s navigation bar 2. Application redirects the user to the profile page 3. User is shown the profile page with: a. Profile Picture b. Username c. Number of Photos d. Number of Reviews e. Number of Likes f. Past reviews
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	7		
Use Case Name:	Add Review of Toilet		
Created By:	Chloie	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered User adds a review to a toilet
Pre-conditions:	1. User is logged into the application
Post-conditions:	1. A review is successfully added to a toilet
Priority:	High
Frequency of Use:	0-1 times a day
Flow of Events:	1. User selects the “Add Review” button on the application’s navigation bar 2. Application directs the user to the Add review page 3. Add review page displays: a. Number of stars (1-5) b. Textbox (300 words or less) c. Upload Photos d. Add Location 4. User clicks the “Add Location” button 5. Application directs the user to the location search page 6. User enters a keyword into the search bar and selects an existing toilet 7. User enters their review in the textbox and chooses a rating 8. Optional: User clicks the “Upload Photos” button and selects photo(s) of the toilet 9. User selects the “Save” button 10. System uploads the new review into the database 11. Application redirects the user to the toilet details page of the chosen toilet
Alternative Flows:	7-AF-1: Add review for new toilet 1. User inputs the address of a new toilet 2. Application displays “Current location does not have a listing yet.” and “Add New Location button” 3. User clicks the “Add New Location” button 4. System returns to step 7 7-AF-2: Add review of existing toilet from “Toilet Details” page

	1. User selects the review button in the “Toilet Details” page 2. Application directs the user to the Add review page 3. Add review page displays: a. Number of stars (1-5) b. Textbox (300 words or less) c. Upload Photos d. Cancel 4. System returns to step 7
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	Review team will evaluate the validity of the user’s request to add a new toilet
Notes and Issues:	-

Use Case ID:	8		
Use Case Name:	View Nearby Toilets		
Created By:	Rachel	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user can view toilets within 500m radius of their current location
Pre-conditions:	1. User is logged into the application 2. User turns on location function and permission is allowed for application to use their current location
Post-conditions:	1. User views a list of toilets within 500m radius of their current location
Priority:	High
Frequency of Use:	0 - 3 times per day
Flow of Events:	1. User selects “Home” button on the application’s navigation bar 2. Application redirects the user to Home page showing the list of toilets near the user 3. Home page displays a list of toilets within 500m radius of the user’s current location a. Location of the toilet b. Distance of nearby toilets from current location c. Toilet Rating d. Type of toilet e. District of toilet
Alternative Flows:	8-AF-1: No nearby toilets within 500m radius of user current location 1. Application displays the message “No toilets nearby. ”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	9		
Use Case Name:	View Toilet Details		
Created By:	J'sen	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user can view additional details of toilets
Pre-conditions:	1. User is logged into the application 2. User is on the home page, map page, favourites page or profile page
Post-conditions:	1. User views the specific details of a toilet
Priority:	Medium
Frequency of Use:	0 - 3 times per day
Flow of Events:	1. User clicks on the specific toilet they want to view the details of 2. Toilet detail page displays the following details: a. Toilet name b. Toilet address c. Toilet crowd level d. Type of toilet (Gender, Handicap, Child-friendly/Nursing room) e. Rating of toilet (Number of stars) f. Reviews of toilet
Alternative Flows:	9-AF-1: User is on the profile page 1. User clicks a past review that they have made 2. System returns to step 2 9-AF-1 : User is on the map page 1. User clicks on a specific toilet pin on the map 2. System returns to step 2 9-AF-1 : User is on the home page 1. User clicks on a specific toilet in the home page 2. System returns to step 2 9-AF-1 : User is on the favourites page

	1. User clicks on a specific toilet that they have favorited 2. System returns to step 2
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	10		
Use Case Name:	View Toilet Gallery		
Created By:	Rachel	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user can view on the image(s) of the selected toilet
Pre-conditions:	1. User is on the “Toilet Details” page
Post-conditions:	1. User views the image gallery of the selected toilet
Priority:	Medium
Frequency of Use:	0 - 3 times per day
Flow of Events:	1. User selects the “Image gallery” button 2. Application directs user to the selected toilet’s gallery page with a grid view of the toilet images
Alternative Flows:	10-AF-1: No images found for selected toilet 1. Toilet gallery page displays “No images found.”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	11		
Use Case Name:	Filter Toilets		
Created By:	Ryan	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user can filter toilets based on a. Type of toilet (Handicap/Gender/Child-friendly/Nursing Room) b. District (Central, North East, North West, South East, South West) c. Distance d. Rating(up to 5 stars)
Pre-conditions:	1. User is logged into the application 2. User has a list of toilets displayed on the map page
Post-conditions:	1. User views the filtered list of toilets
Priority:	Low
Frequency of Use:	0 - 3 times per day
Flow of Events:	1. User clicks on the specific filter(s) they would like to apply a. Search Keyword b. Type c. District d. Distance e. Rating 2. Application displays the updated list of toilets after applying filters selected
Alternative Flows:	11-AF-1: If no toilets are found after applying filter(s) 1. The application displays “No toilets nearby.”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	12		
Use Case Name:	Edit review		
Created By:	Ryan	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user can edit reviews that they have written
Pre-conditions:	1. User is logged into the application 2. User is in the “Toilet details” page
Post-conditions:	2. User successfully edited the review that they have written
Priority:	Low
Frequency of Use:	0 - 1 times every 3 months
Flow of Events:	1. User clicks the “Edit Review” button 2. Application redirects the user to “Edit review” page 3. Application then display the following: a. Rating (1 - 5 stars) b. Textbox with their review c. Attached images d. Location of toilet 4. User can select the textbox to edit their comment 5. User can add additional images 6. User can change the review rating 7. User selects the “Save” button to save all changes made 8. System updates the review details in the database 9. Application redirects the user to the toilet details page
Alternative Flows:	-
Exceptions:	12-EX-1 : User decides not to make changes to their review, the user has the option to return back by selecting the “Back” button
Includes:	-
Special Requirements:	-
Assumptions:	User has attached images to their review
Notes and Issues:	-

Use Case ID:	13		
Use Case Name:	Delete review		
Created By:	Ryan	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user can delete reviews that they have written
Pre-conditions:	1. User is logged into the application 2. User is in “Toilet Details” page
Post-conditions:	1. User successfully deleted the review that they have written
Priority:	Low
Frequency of Use:	0 - 1 times per month
Flow of Events:	1. User clicks the “Edit Review” button 2. Application redirects the user to “Edit review” page 3. User selects the “Delete” button to delete this review 4. Application displays a “Confirm Delete” prompt 5. User selects the “Confirm” button 6. System deletes the review
Alternative Flows:	-
Exceptions:	13-EX-1 : In the “Edit Review” page, if the user decides to make changes to their review instead of deleting it, the user has the option to do so in the same page 13-EX-2 : In the “Edit Review” page, if the user decides not to delete the review, user has the option to return back to the “Toilet Details” page by selecting the “Back” button 13-EX-3 : In the “Confirm Delete” prompt, if the user decides not to delete the review, user has the option to return back to the “Edit Review” page by selecting the “Cancel” button
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	14		
Use Case Name:	Search for toilets		
Created By:	Ryan	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user can search for toilets near desired destination / current location
Pre-conditions:	1. User is logged into the application 2. User is on the “Map” page
Post-conditions:	1. Application displays list of toilets near specified location
Priority:	High
Frequency of Use:	1 - 3 times per day
Flow of Events:	1. User enters location into search bar 2. User clicks on a suggested location 3. Moves camera to user’s specified location 4. Map page displays markers of toilets near the specified location
Alternative Flows:	-
Exceptions:	14-EX-1: Searched location does not exist
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	15		
Use Case Name:	View Directions to Toilet		
Created By:	Rachel	**Last Updated By:**	-
Date Created:	9 September	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Provide navigation guidance to user using google maps
Pre-conditions:	1. User is on the “Toilet Details” page
Post-conditions:	1. Application redirects user to Google Maps
Priority:	High
Frequency of Use:	0 - 3 times per day
Flow of Events:	1. User clicks on “Directions” button 2. System redirects user to Google Maps with destination set as toilet location
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	1. User has google maps installed in their device
Notes and Issues:	-

Use Case ID:	16		
Use Case Name:	Add Toilet to Favourites		
Created By:	Deuel	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

Actor:	Registered User (Initiating Actor)
Description:	Registered user can add a toilet to favourites
Pre-conditions:	1. User is logged into the application 2. User is at the “Toilet Details” page
Post-conditions:	1. User successfully adds toilet to the list of favourite toilets
Priority:	Medium
Frequency of Use:	0-3 times per day
Flow of Events:	1. User clicks on the “Favourites” button on the Toilet Detail Page 2. Application adds the toilet to the list of favourites
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	17		
Use Case Name:	Remove Toilet from Favourites		
Created By:	Deuel	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

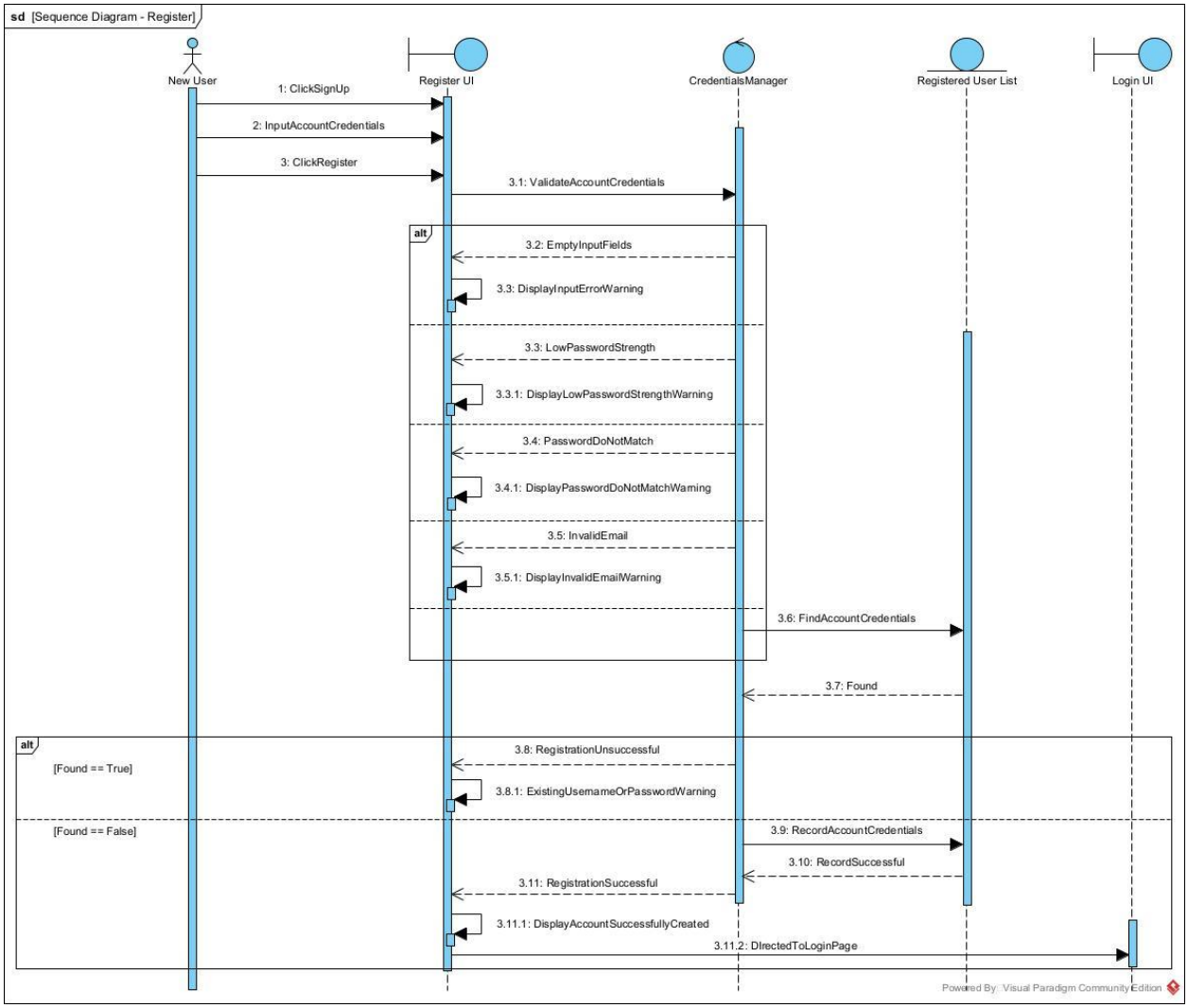
Actor:	Registered User (Initiating Actor)
Description:	Registered user can remove a toilet from favourites
Pre-conditions:	1. User is logged into the application
Post-conditions:	1. User successfully removes toilet from their list of favourite toilets
Priority:	Medium
Frequency of Use:	0 - 3 times per day
Flow of Events:	1. User clicks on the “Favourites” button on the application navigation bar 2. Application displays “Favourites” page which has a list of user’s favourites toilets 3. User taps on “Heart” button to remove toilet from “Favourites” page 4. Application removes specified toilet from “Favourites” page
Alternative Flows:	17-AF-1: User is on the “Toilet Details” page 1. User clicks on the “Favourites” button on the Toilet Detail Page 2. System returns to step 4
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	1. There are existing toilets in the favourites list.
Notes and Issues:	-

Use Case ID:	18		
Use Case Name:	View Favourite Toilets		
Created By:	Deuel	**Last Updated By:**	-
Date Created:	9 September 2023	**Date Last Updated:**	-

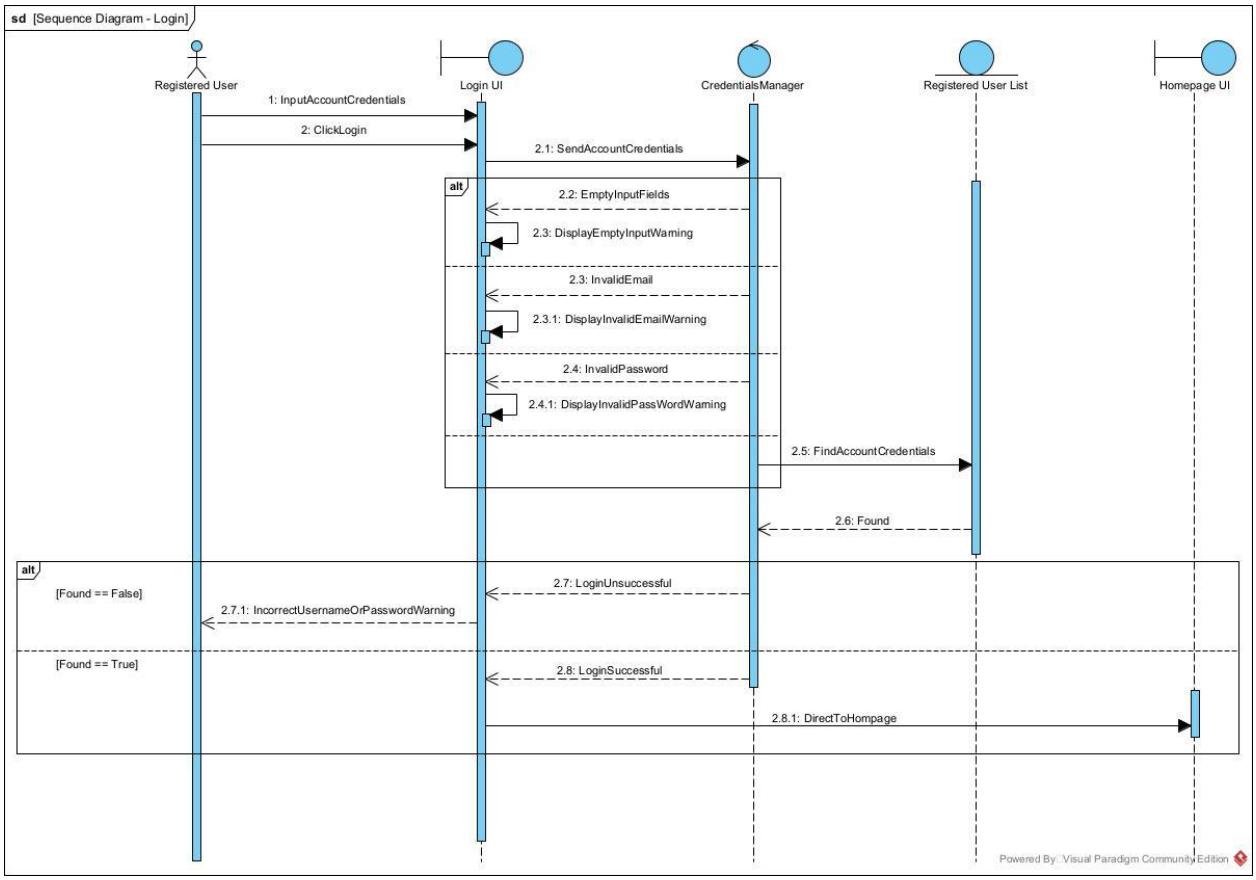
Actor:	Registered User (Initiating Actor)
Description:	Registered user can view their list of favourite toilets
Pre-conditions:	1. User is logged into the application 2. User has a toilet added to their list of favourites
Post-conditions:	1. User successfully views their list of favourite toilets
Priority:	Medium
Frequency of Use:	0 - 2 times per day
Flow of Events:	1. User clicks on the “Favourites” button on the application navigation bar 2. Application displays a list of the user’s favourites toilets
Alternative Flows:	18-AF-2: User has no favourite toilets 1. Application displays “No Favourite Toilets”.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

4.2 Sequence Diagrams

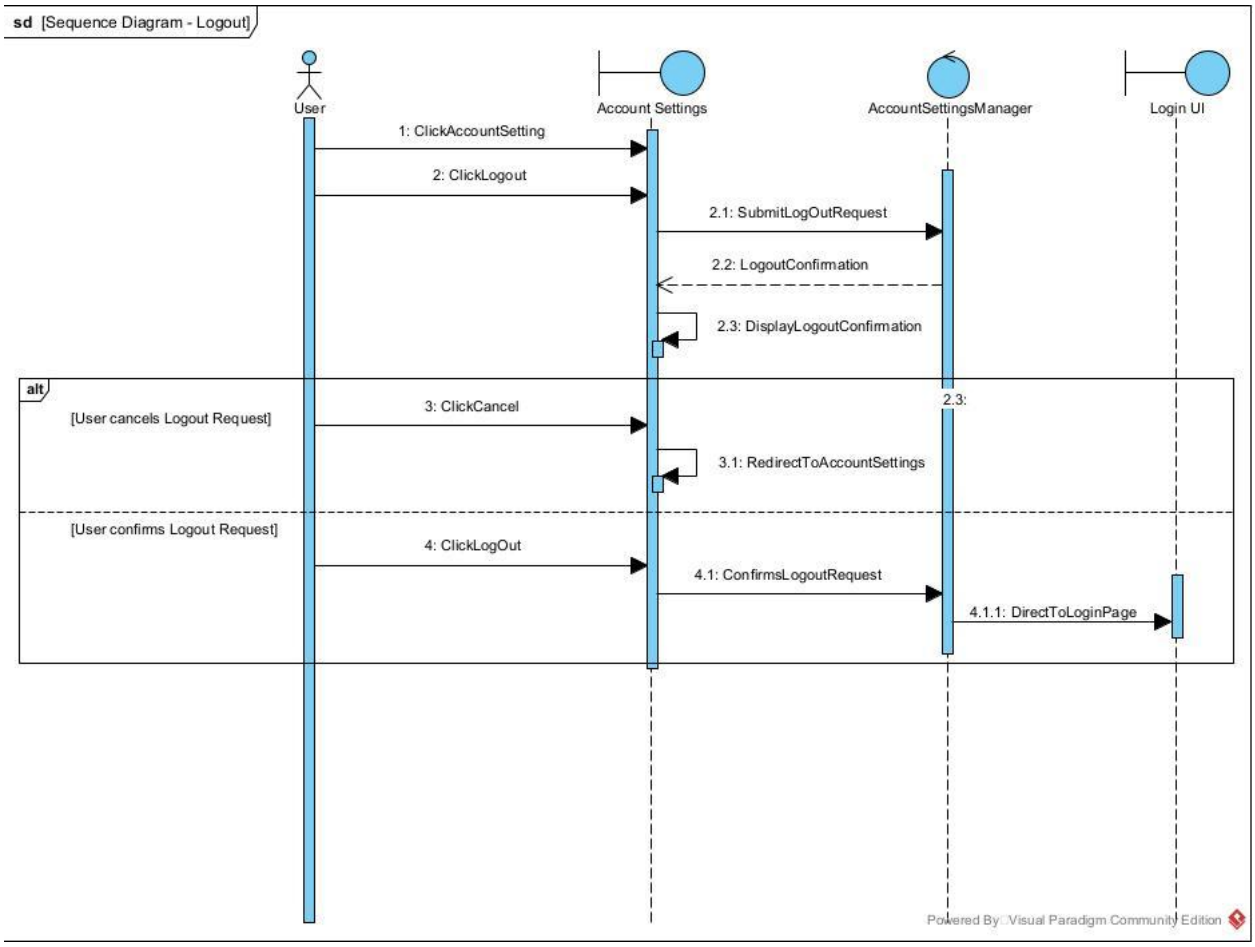
Use Case 1: Register


sd [Sequence Diagram - Register]
New User
Register UI
CredentialsManager
Registered User List
Login UI
1: ClickSignUp
2: InputAccountCredentials
3: ClickRegister
3.1: ValidateAccountCredentials
alt
3.2: EmptyInputFields
3.3: DisplayInputErrorWarning
3.3: LowPasswordStrength
3.3.1: DisplayLowPasswordStrengthWarning
3.4: PasswordDoNotMatch
3.4.1: DisplayPasswordDoNotMatchWarning
3.5: InvalidEmail
3.5.1: DisplayInvalidEmailWarning
3.6: FindAccountCredentials
3.7: Found
alt
[Found == True]
3.8: RegistrationUnsuccessful
3.8.1: ExistingUsernameOrPasswordWarning
[Found == False]
3.9: RecordAccountCredentials
3.10: RecordSuccessful
3.11: RegistrationSuccessful
3.11.1: DisplayAccountSuccessfullyCreated
3.11.2: DIrectedToLoginPage
Powered By: Visual Paradigm Community Edition

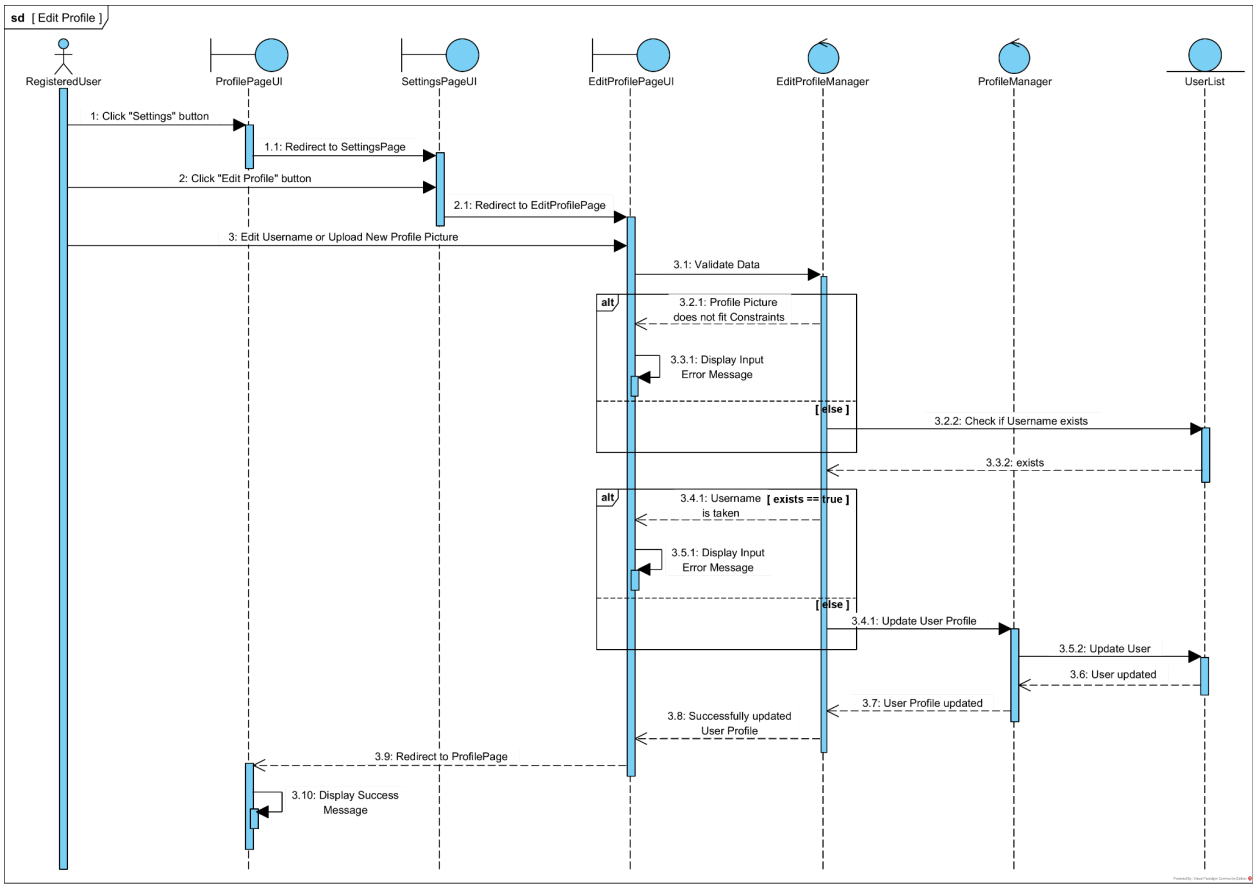
Use Case 2: Login



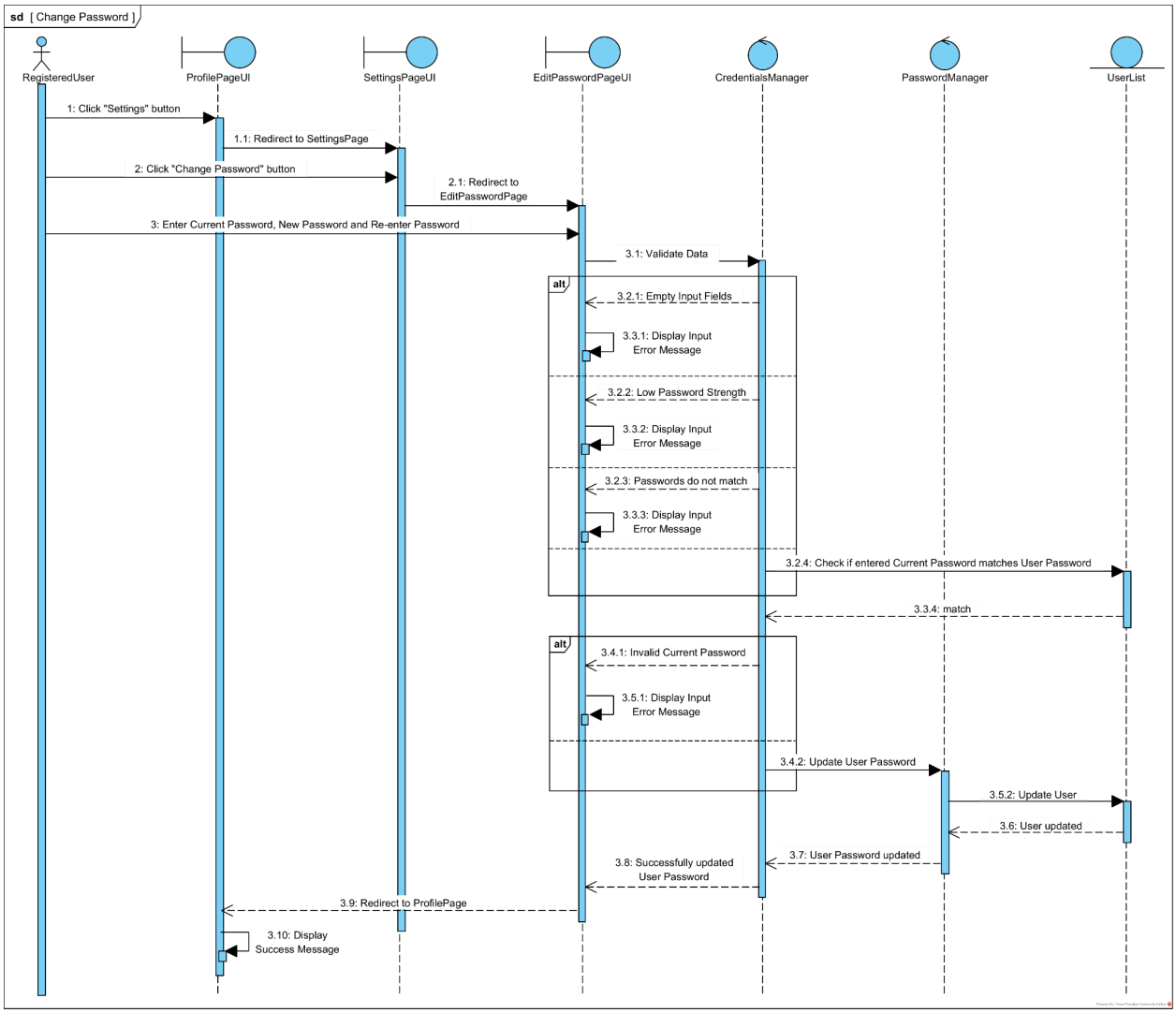
Use Case 3: Logout



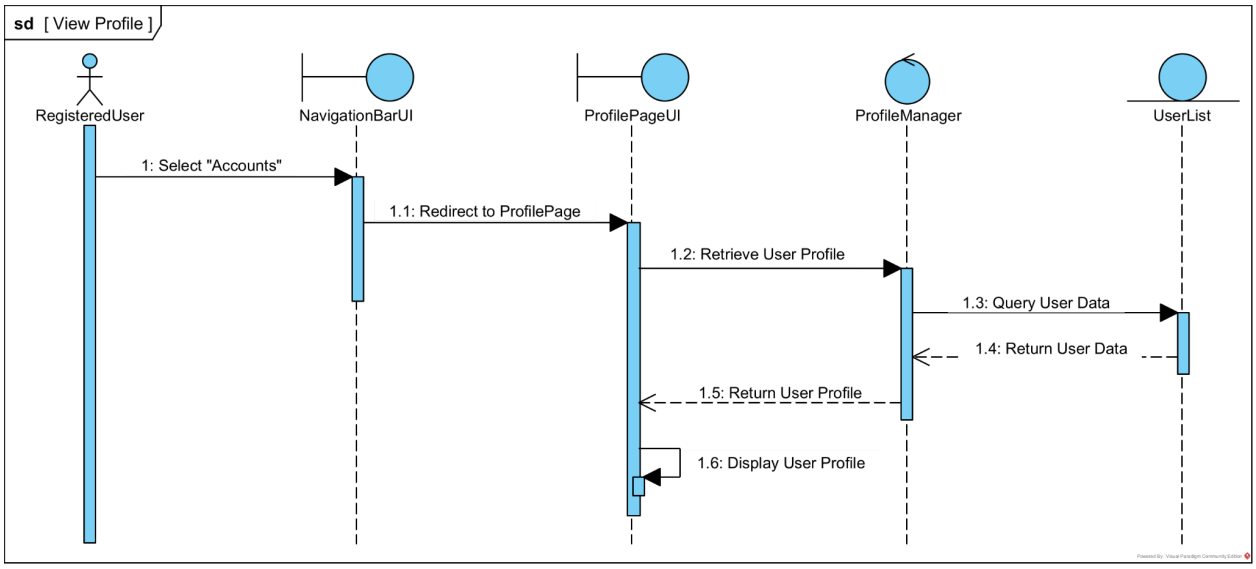
Use Case 4: Edit Profile



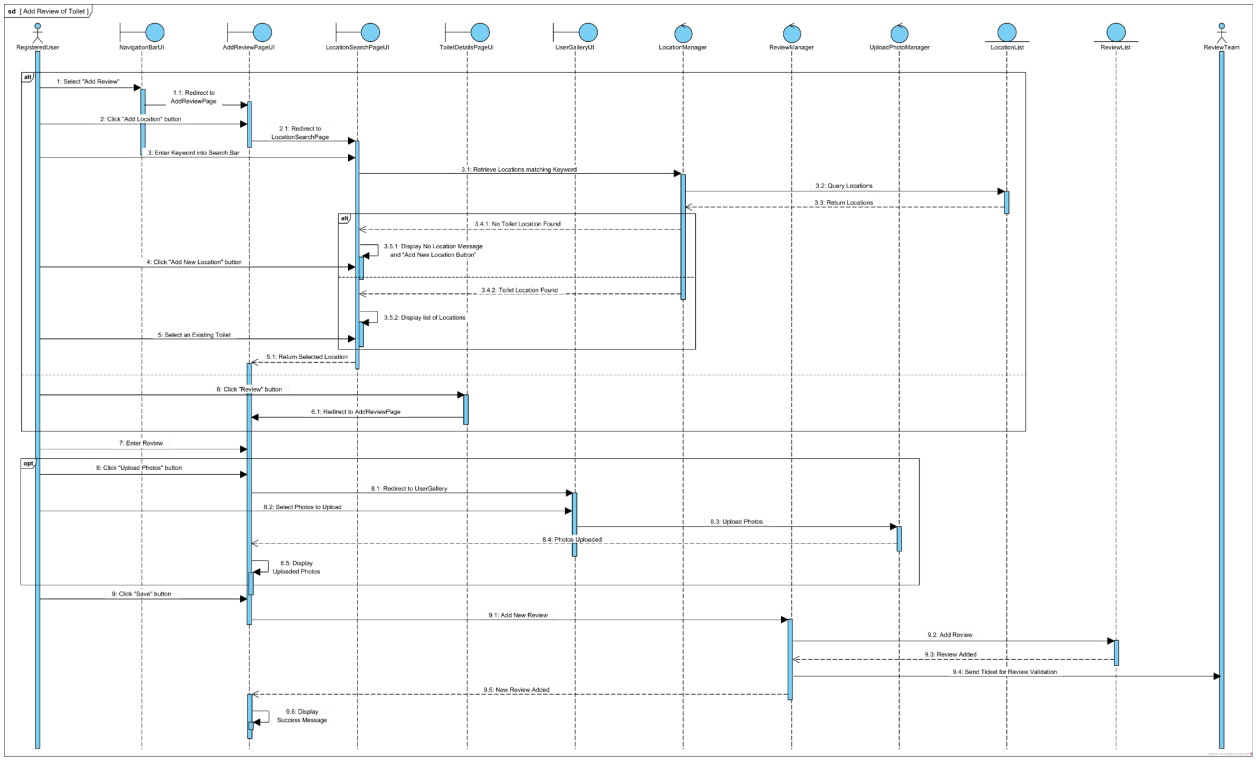
Use Case 5: Change Password



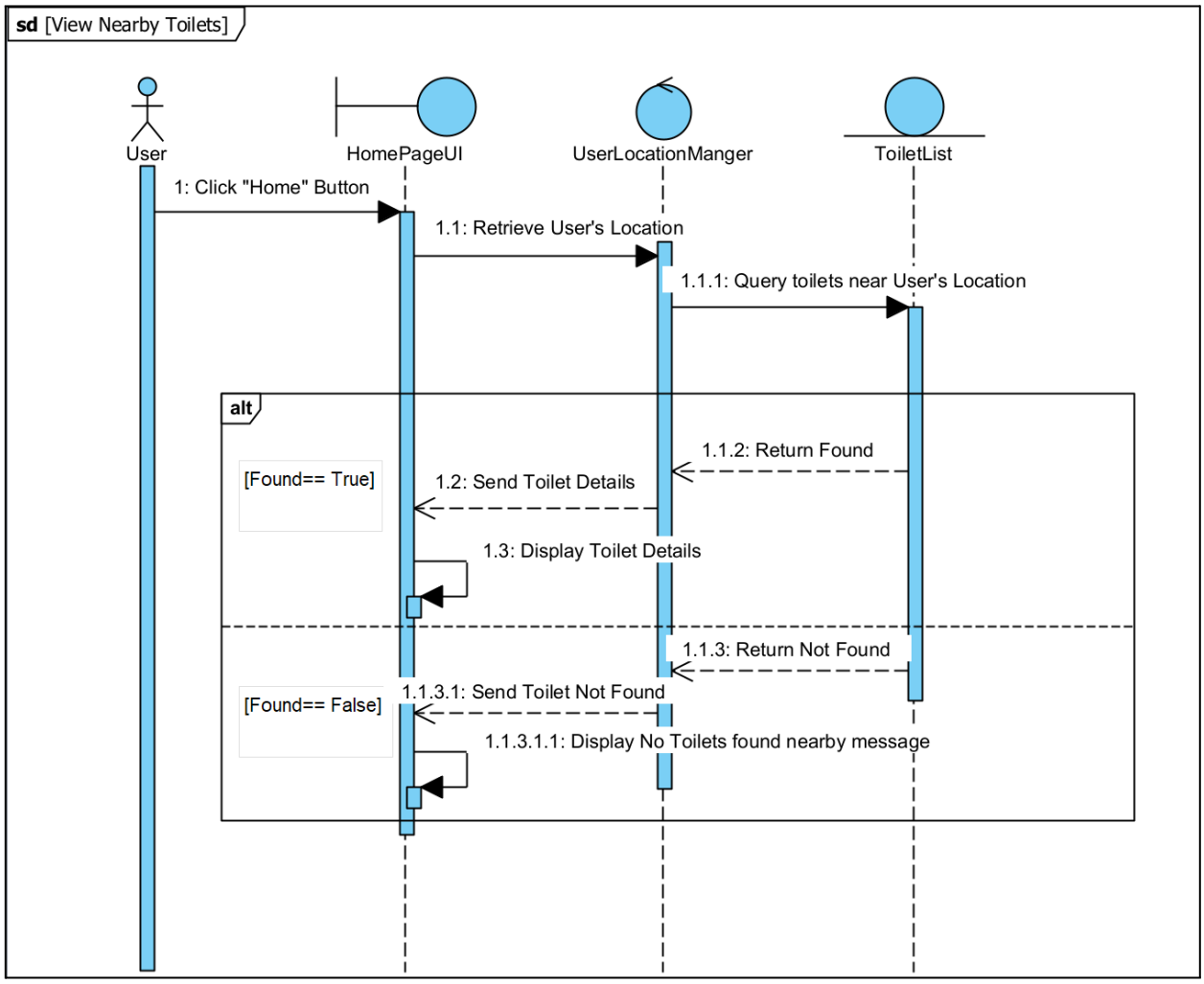
Use Case 6: View Profile



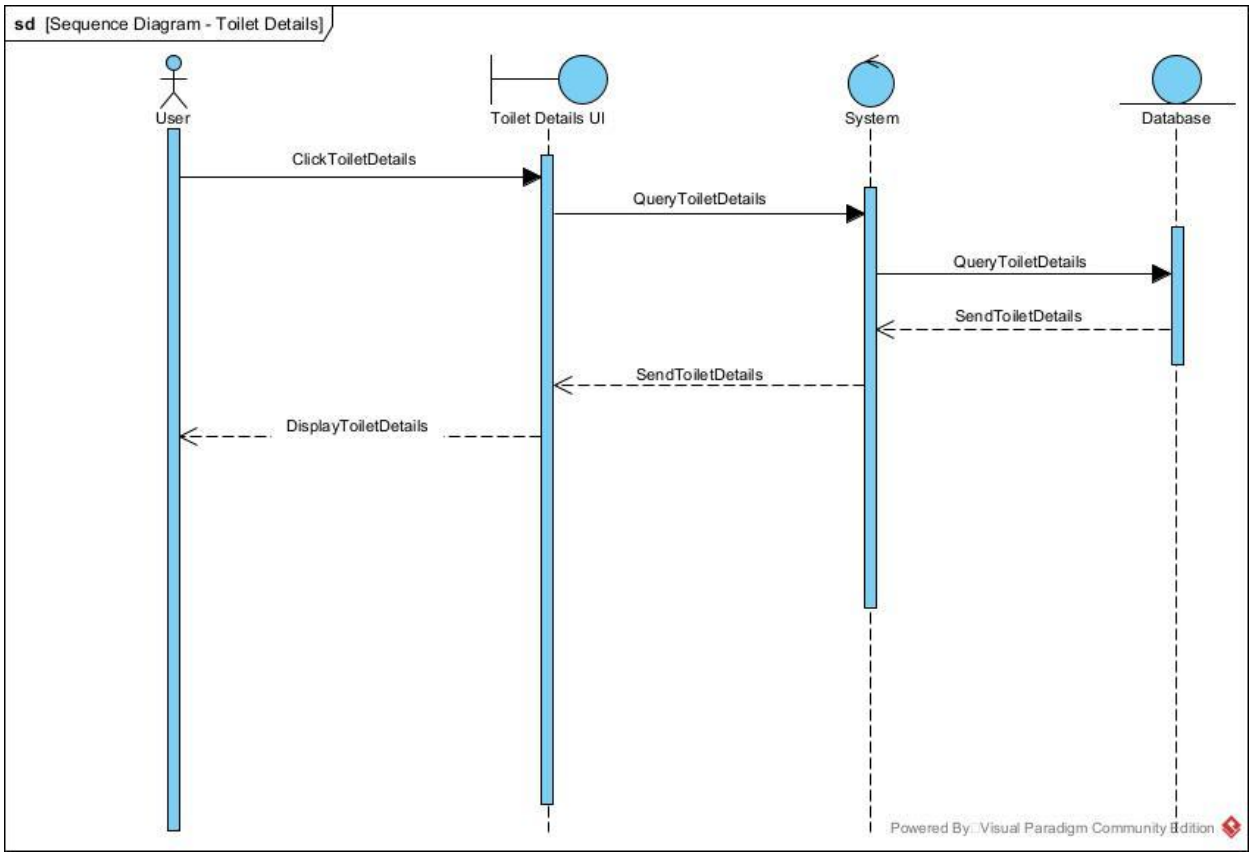
Use Case 7: Add Review of Toilet



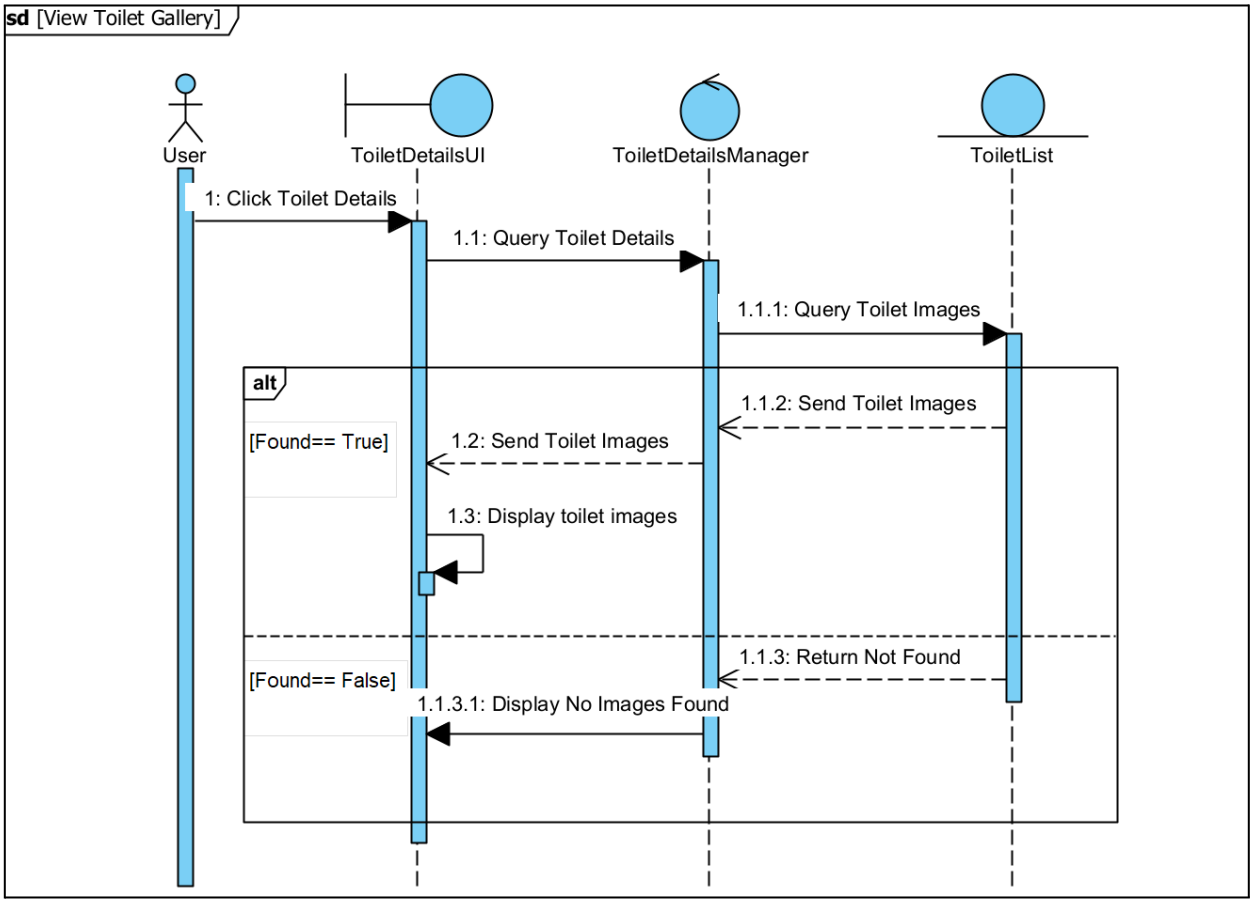
Use Case 8: View Nearby Toilets



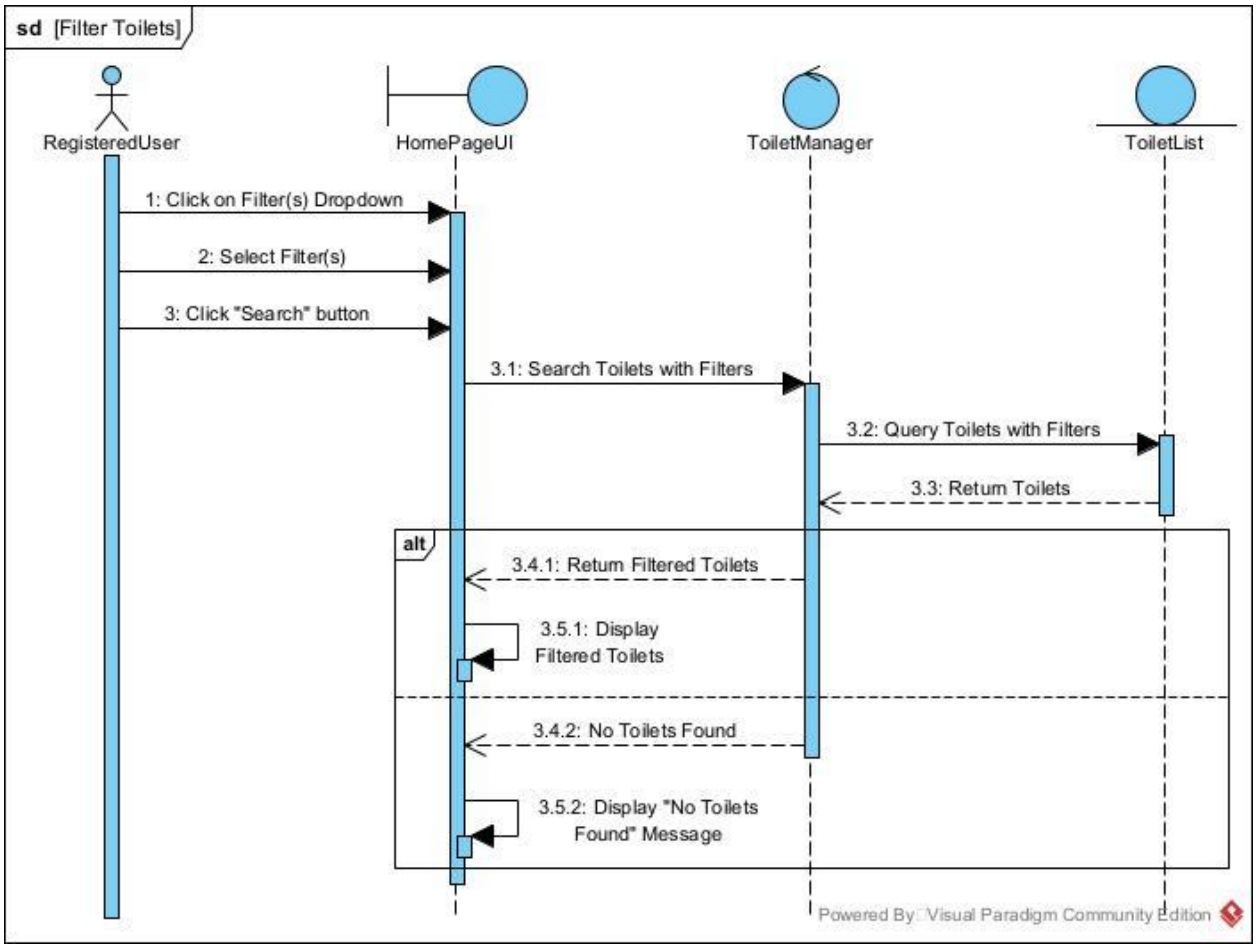
Use case 9: Toilet Details



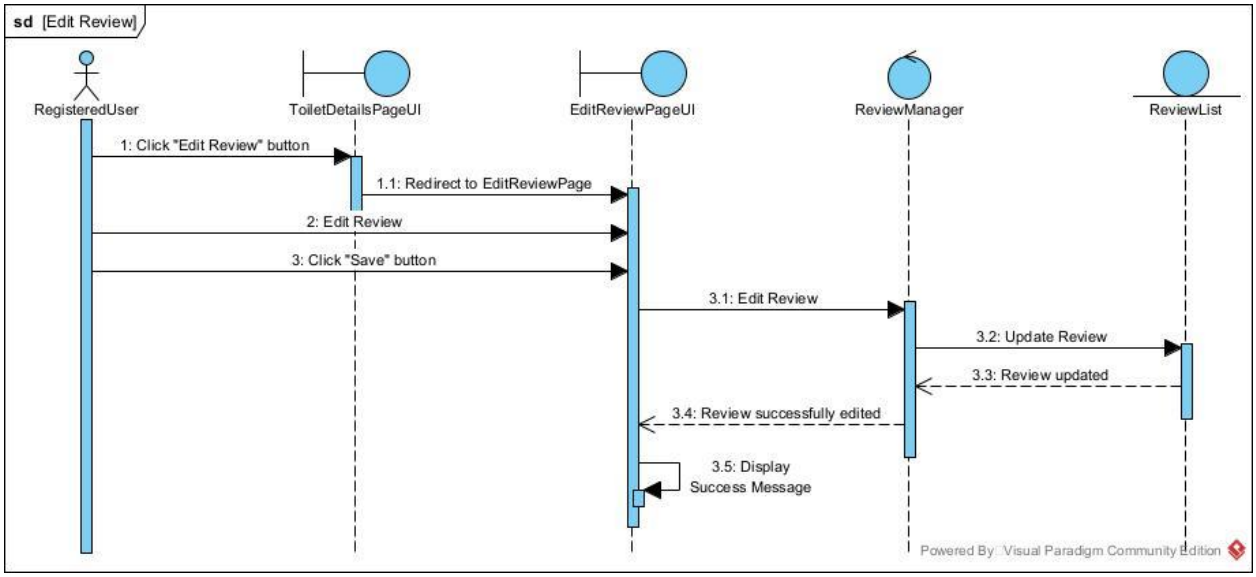
Use case 10: View Toilet Gallery



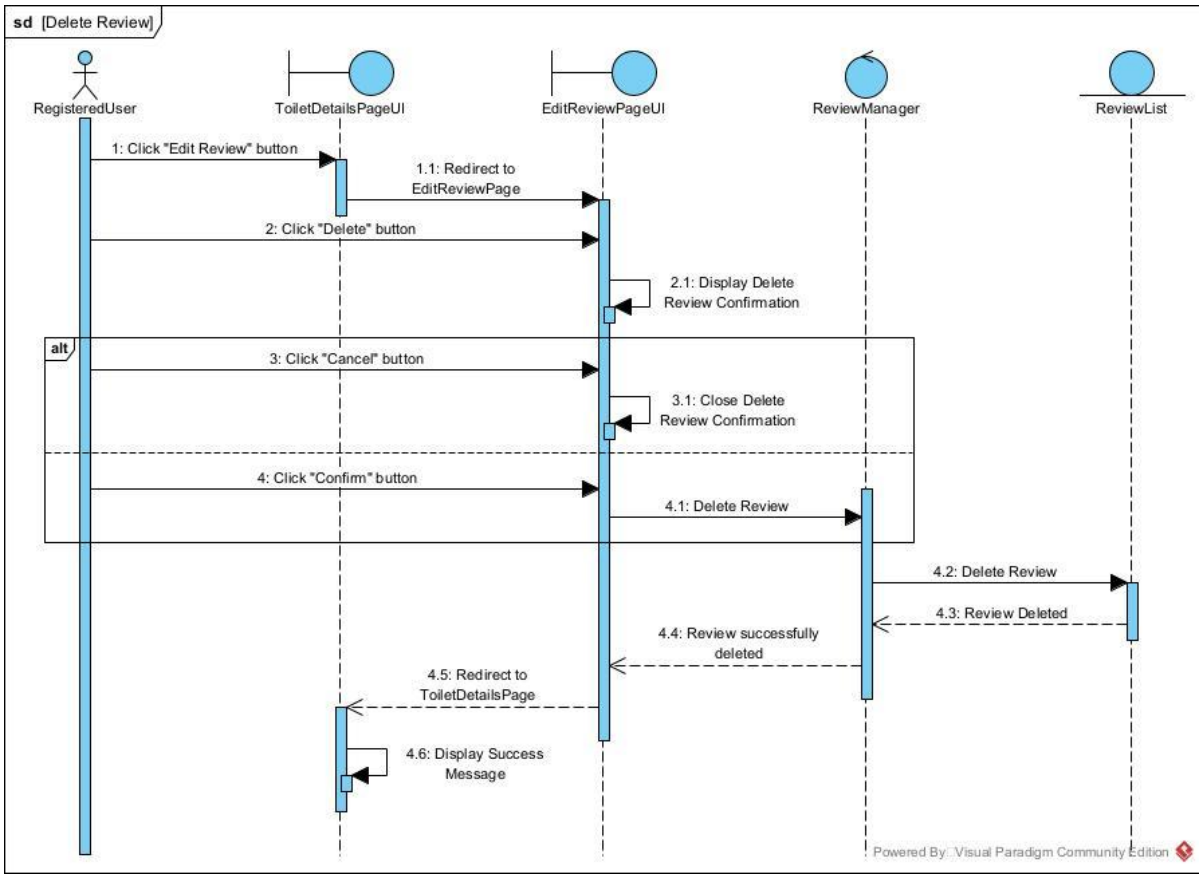
Use case 11: Filter Toilet



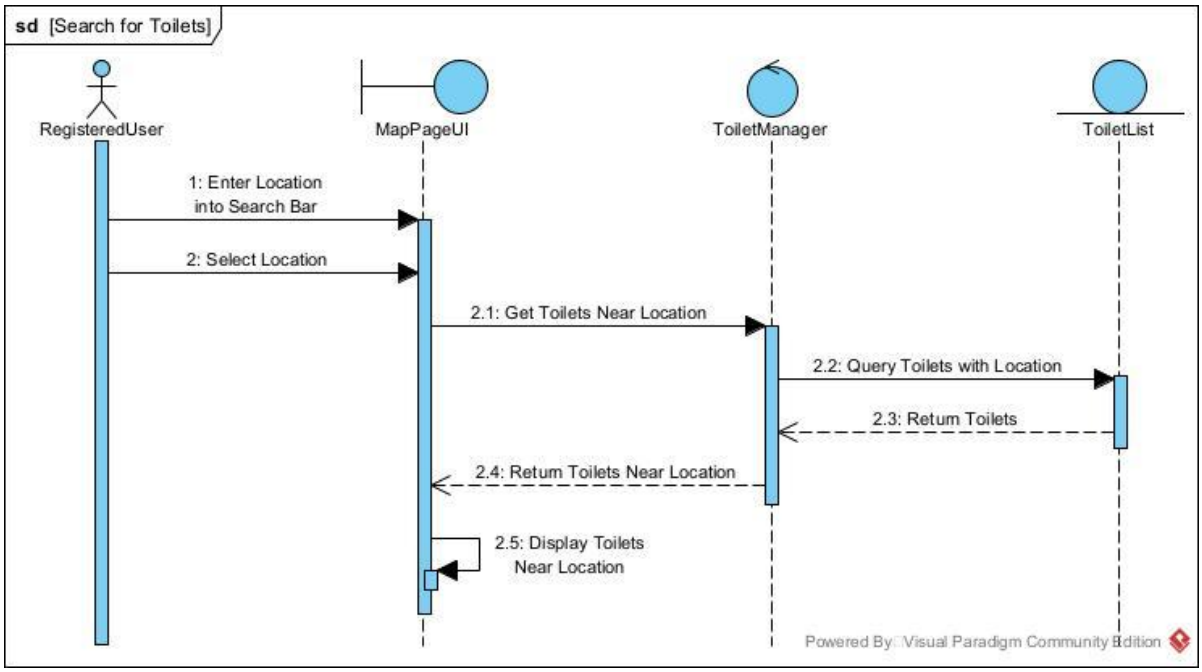
Use case 12: Edit review



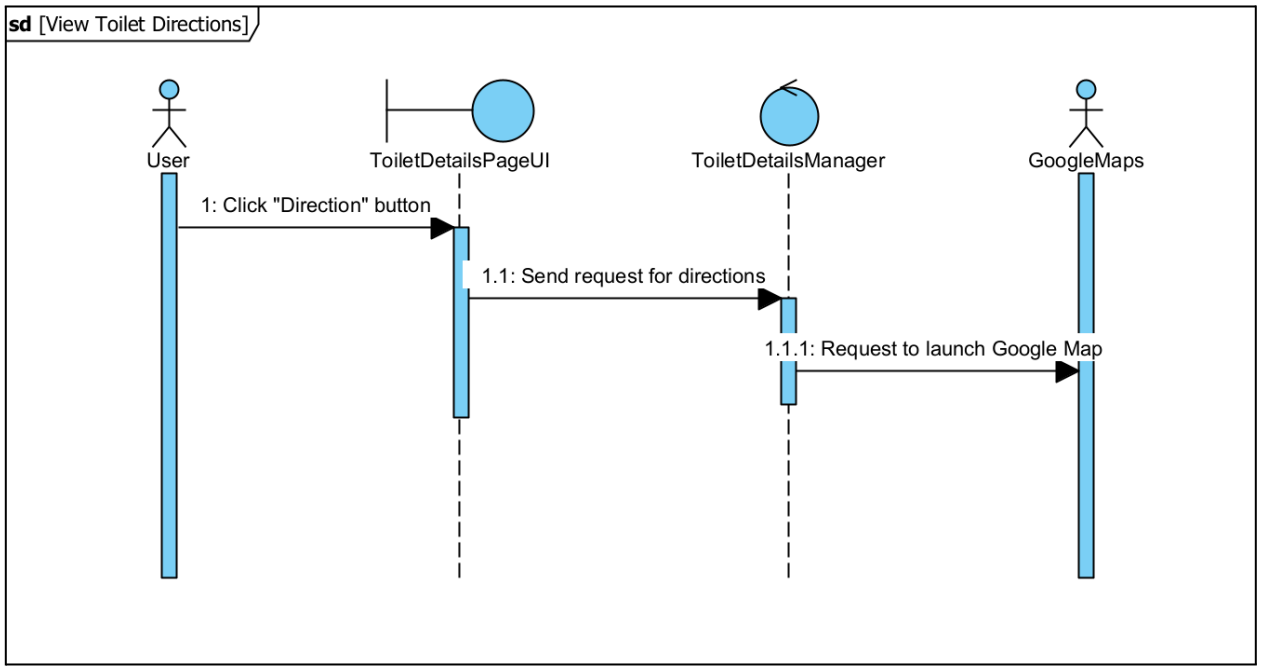
Use case 13:Delete Review



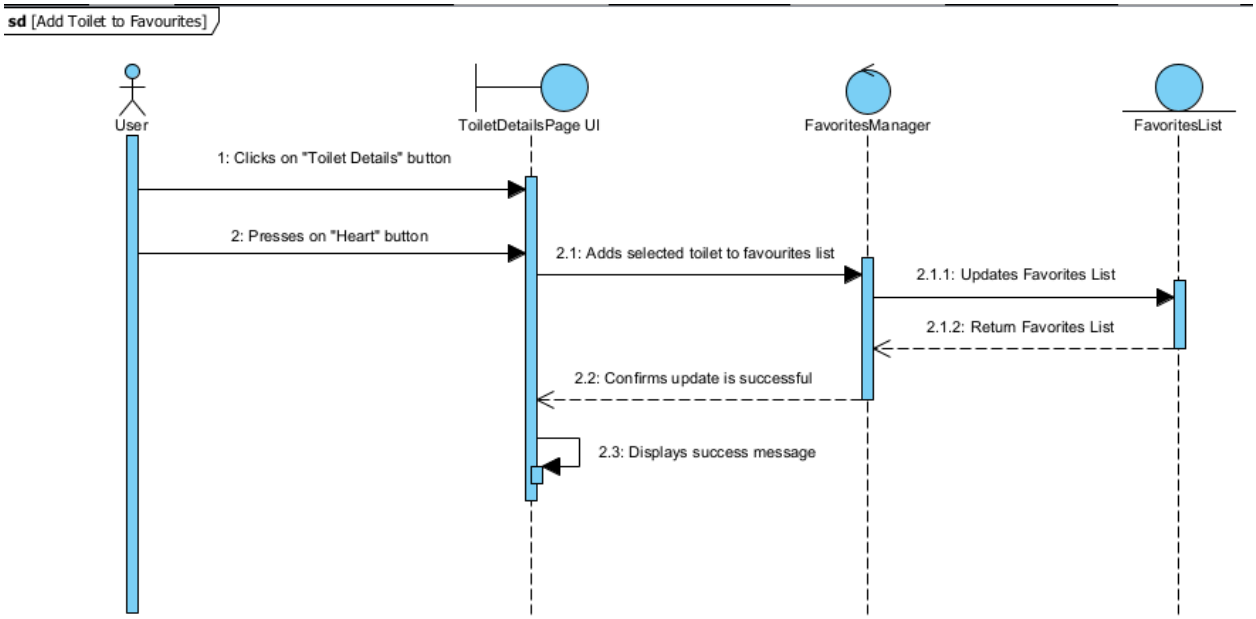
Use case 14:Search for toilet



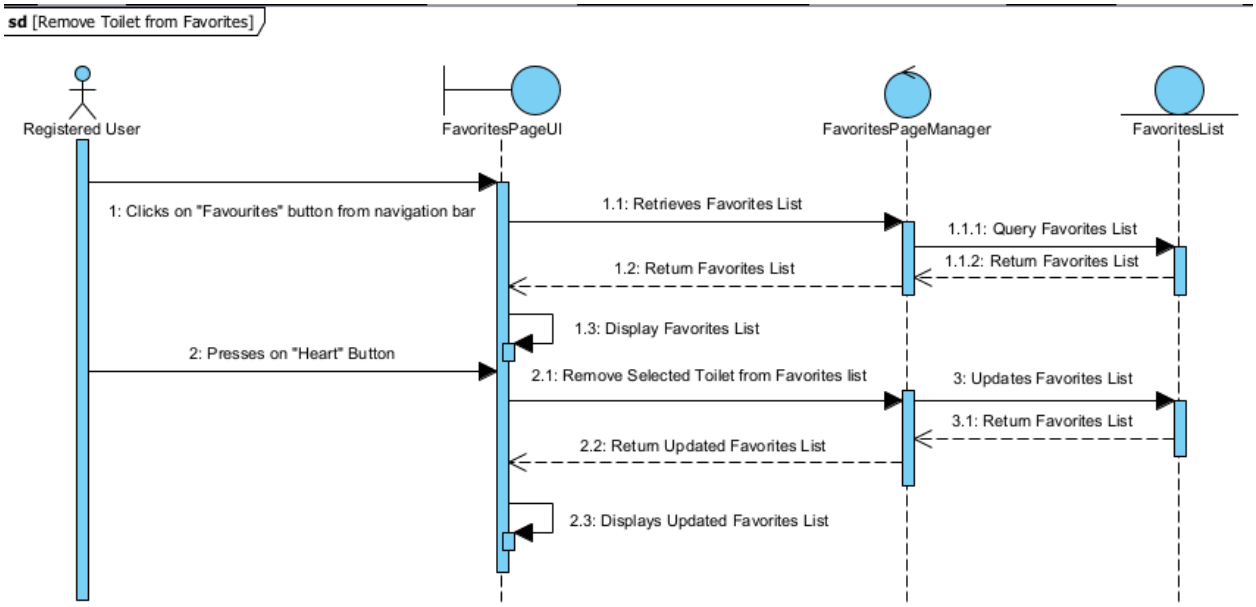
Use case 15: View Directions to Toilet



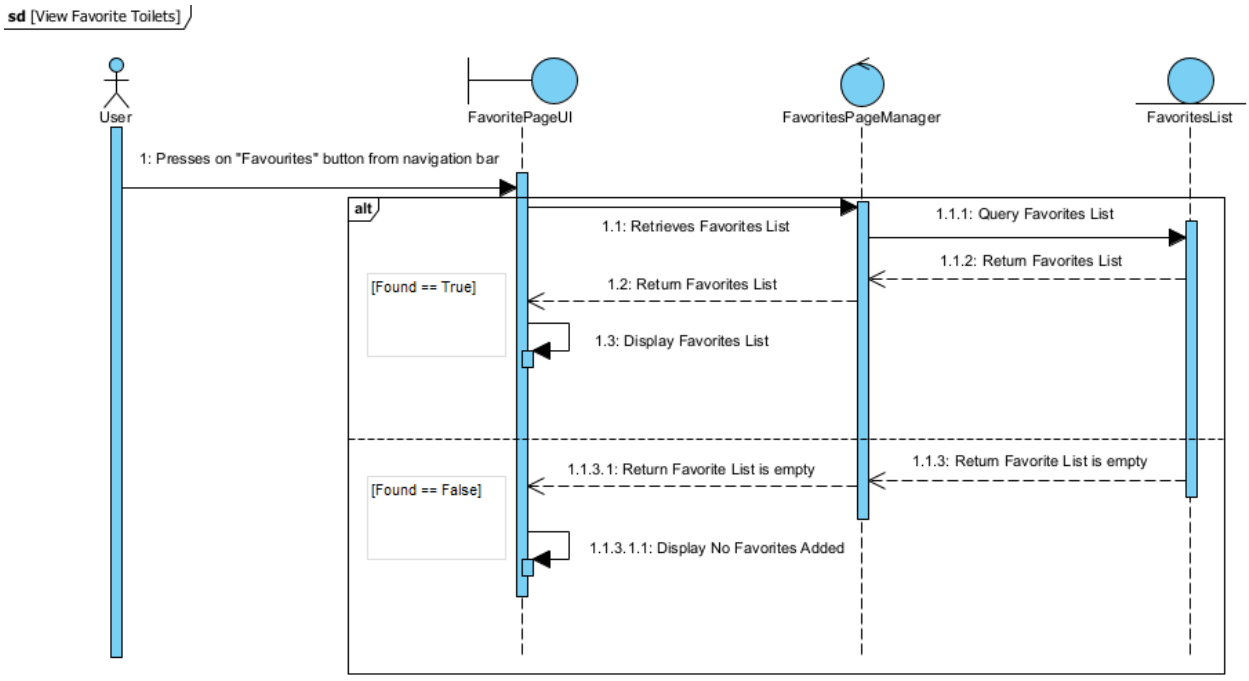
Use case 16: Add Toilet to Favourites



Use case 17: Remove Toilet from Favourites



Use case 18: View Favourite Toilets



5. Other Nonfunctional Requirements

5.1 Performance Requirements

Our application must display swift response times and efficient data retrieval, particularly in the context of dynamic searches for nearby toilets. Real-time features, like the integration of Google Maps, Google Places API and LTA APIs, consistent and unhindered operations to provide users with timely and accurate information. As the user base expands, the Firebase database used by our application's backend infrastructure must scale to manage increased data loads without compromising performance. Additionally, to prevent user frustration, all functionalities should be functioning within 5 seconds.

5.2 Security Requirements

Our application uses Firebase Authentication and Firestore database which places an emphasis on security. In addition, Firebase, as a database platform, is certified under major privacy and security standards. The Firebase Authentication requires secure user identity verification and data access controls within Firestore will be strictly limited to authorised users, ensuring user-specific data remains only exclusively accessible to the authenticated owner.

To maintain data integrity, "LooLah" will undergo periodic security audits and vulnerability assessments, ensuring stringent adherence to data protection regulations, such as PDPA (Personal Data Protection Act) which will reinforce the application's commitment to user-centric security practices.

5.3 Software Quality Attributes

Our application is developed with user satisfaction and developer efficiency in mind. Thus, the following must be taken into consideration:

- **Usability**: Ensuring an intuitive and user-friendly interface, making navigation and interaction fuss-free for users.
- **Reliability**: Consistent performance and data accuracy, ensuring users can depend on "LooLah" for accurate and timely information about nearby toilets.
- **Maintainability**: The ease with which the application can be updated, enhanced, and maintained over time without disrupting user experience.
- **Flexibility**: Allowing for the incorporation of new features and functionalities without causing significant disruptions to the existing system.
- **Robustness**: The application's resilience in handling unexpected inputs, errors, or adverse conditions without compromising overall functionality.
- **Scalability**: Ensuring that "LooLah" can handle increased data loads and user traffic as the application grows, without sacrificing performance.

5.4 Business Rules

User must be registered to use the application:

Our application requires users to register an account beforehand. Only when they have created an account and logged in using the corresponding account, will they be able to access the functionalities of our application.

Appendix A: Data Dictionary

Term	Definition
Username	An accounts unique identification, can be used to search for user, acts as a link to account's profile
Profile	A page containing user's profile picture and past toilet reviews
Profile Picture	A jpeg file uploaded by user for personalization
Review	300 words or less describing the overall toilet experience
Locate Toilet	A function to aid the user in locating nearby toilets
Navigate	A function to help the user to get to the desired location in the fastest way
Like	A function to assist user to mark selected post, so that similar posts will be recommended to the user
Rating	A classification ranking of the toilet based on the assessment of its cleanliness, quality and overall user satisfaction
District	The general locations where the toilets are found in Singapore (Central, North East, North West, South East, South West)
Toilet Type	The types of toilets that are available (Gender, Handicap, Child-friendly)
Favourite list	A page containing user's favorited toilets
Page	An interface containing corresponding information with regards to the subject(E.g. "Toilet Details" page shows the necessary information of the selected toilet)