Software Requirements Specification

for

Wastetastic

Version 1.0 approved

Prepared by

Siddharth, Agnesh, Khush, Rachita, Kartikeya

Ctrl Alt Defeat

25 January 2021

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Introduction

1.1.Purpose

Wastetastic is an Android and iOS mobile app which allows users across Singapore to view waste disposal vendors/recycling points of 5 different categories: Normal Waste, E-Waste, Lighting Waste, Waste Treatment (Industrial Waste) and Cash for Trash. Its purpose, under the Smart Nation Singapore program, is to incentivize users to dispose off waste responsibly, in times when the environment is in dire need of our help. This document outlines the plan involved in the software development of the Wastetastic App. The intended readers of this document are current and future developers of Wastetastic.

1.2.Document Conventions

This document features the usage of the font Times New Roman of size 12. It must be noted that each new header uses Times New Roman of size 18 and is stylized to bold. Each subheading uses Times New Roman of size 14 and is stylized to bold. In numbering our detailed requirements and use cases we employ a nested (indented) numbering system whereby a title numbered 1 has subheadings 1.1, 1.2 and so on.

1.3.Intended Audience and Reading Suggestions

The Wastetastic App is targeted at all residents of Singapore above the age of 12 years. We believe that our app can help instill a sense of environmental responsibility early on. The app would also be specifically useful to business owners such as factory owners, restaurants owners, etc.

1.4.Product Scope

Wastetastic serves as a guide for users on waste disposal and recycling points (Waste POIs) all across Singapore. It provides information about these various waste POIs in the form of description, address, location, working hours, etc. It further also allows users to check the real time parking availability near a Waste POI, should the user be planning to dispose of his waste immediately. Finally, it also gives users general guidelines on how to recycle and allows users to keep track of the personal waste that they dispose off.

1.5 Future Plans

- Separate user domains can be created for regular users and business owners.
- Time series analysis can be performed to predict the user's future activity.
- A tensorflow lite model with the help of Google Teachable Machine can be implemented to verify that the user has indeed uploaded a picture of a waste.
- A social networking system can be implemented where users can add friends and view other users' activities.

Overall Description

2.1.Product Perspective

With a 7-fold increase in the amount of solid waste disposed of, responsible waste disposal has become an issue of primary importance. As residents of Singapore, it is our prime duty to ensure that we do our part in contributing to this noble cause. Our Wastetastic App aims to provide an integrated platform for managing one's waste, while providing a seamless experience.

2.2.Product Functions

Effectively the Wastetastic App must serve the following 4 main functionalities:

2.3. The first function is to present the nearby waste collectors based on the choice of waste and the user device's GPS location. The App will tell the location of nearby waste POIs speedily and thereby give a fast response to the user.

2.4. The second function is to notify the user about the nearest parking area based on the location of the waste collectors. Due to this feature, the nearby car parking information will be retrieved, and the App can present this information to the user efficiently.

2.5. The third function is to present past wastage disposal data analytics to the user. This will enable the user to keep track of the amount of waste disposed of.

2.6. The fourth function will be to display general information and guidelines on recycling. This will educate the user about recycling waste and therefore encourage as well as guide the user on disposing of waste in a proper manner.

2.7. Operating Environment

This application will run on all Android as well as iOS devices.

2.8. Design and Implementation Constraints

Firebase has been selected to be used as the database of the application. Wastetastic is not affiliated with Firebase in any way. The developers of this application are not liable for any risks associated with storing user data on Firebase. Any attempts on hacking the database or any data breach must not be traced back to the developers.
This application is currently only available in English.

2.9. User Documentation

The website is designed to be intuitive and user friendly. No other documentations are currently available.

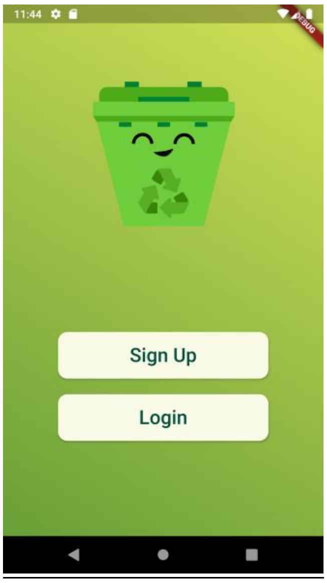
2.10. Assumptions and Dependencies

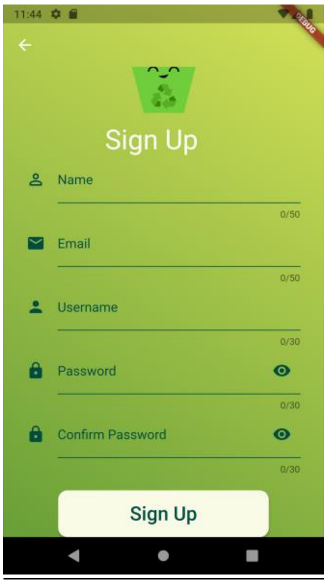
The carpark availability is determined using the data obtained from the API from https://api.data.gov.sg/v1/transport/carpark-availability. The reliability and accuracy of the recommendations depends on the called API.
The Google Map of Wastetastic is based on the Google Maps API from https://cloud.google.com/maps-platform/maps. This is used to get the user input of location and is used as part of our recommendation system.

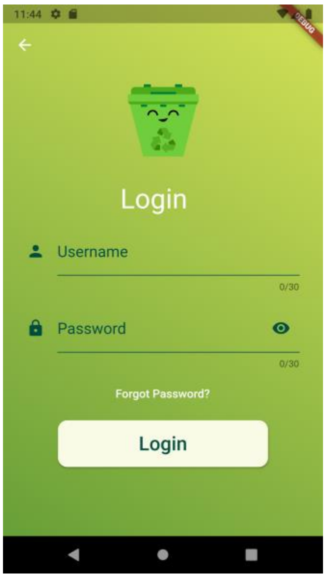
External Interface Requirements

3.1.User Interfaces

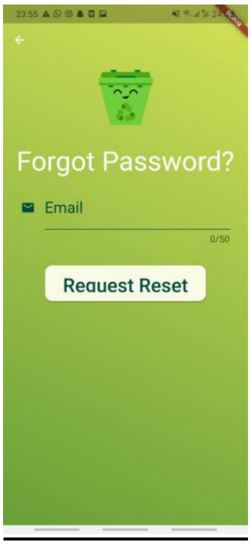
3.1.1. Login and Registration screens:



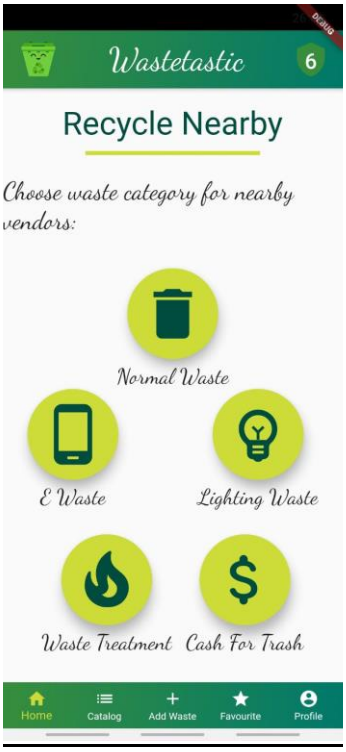




3.1.2. Forgot Password Screen:

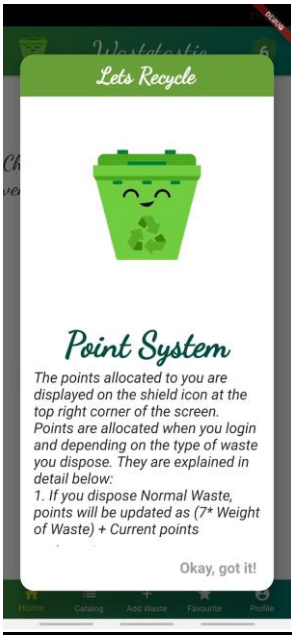


3.1.3. Home Screen:



Choose waste category for nearby vendors:

3.1.4. General Information:

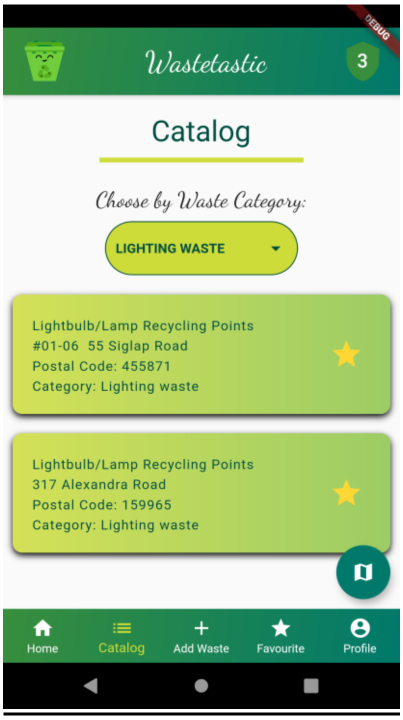


Point System

The points allocated to you are displayed on the shield icon at the top right corner of the screen. Points are allocated when you login and depending on the type of waste you dispose. They are explained in detail below:

1. If you dispose Normal Waste, points will be updated as (7* Weight of Waste) + Current points

3.1.5. Catalog Screen:



Catalog

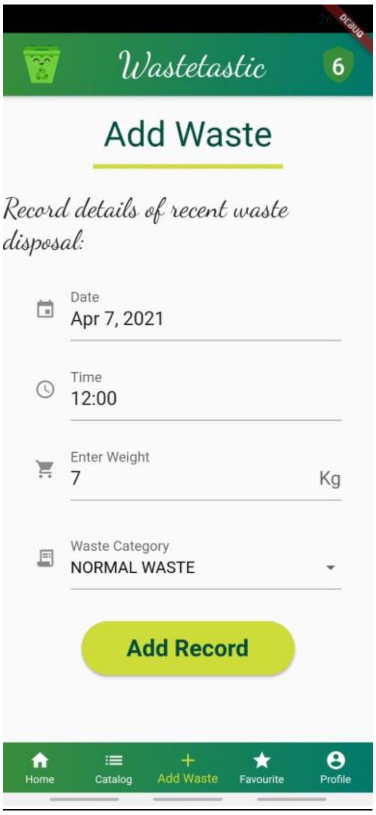
Choose by Waste Category:

LIGHTING WASTE

Lightbulb/Lamp Recycling Points
#01-06 55 Siglap Road
Postal Code: 455871
Category: Lighting waste

Lightbulb/Lamp Recycling Points
317 Alexandra Road
Postal Code: 159965
Category: Lighting waste

3.1.6. Add New Waste Record Screen:



Add Waste

Record details of recent waste disposal:

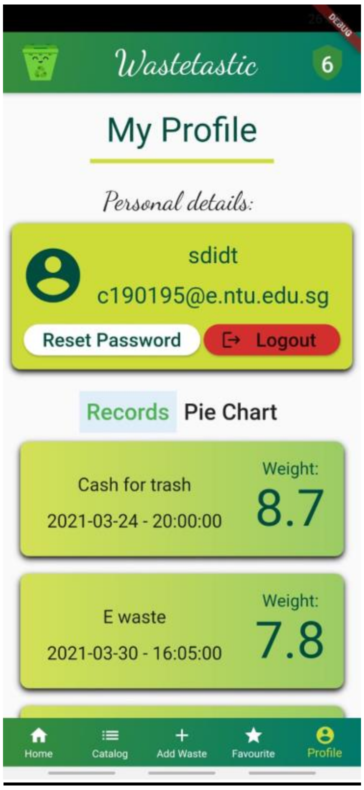
Date
Apr 7, 2021

Time
12:00

Enter Weight
7 Kg

Waste Category
NORMAL WASTE

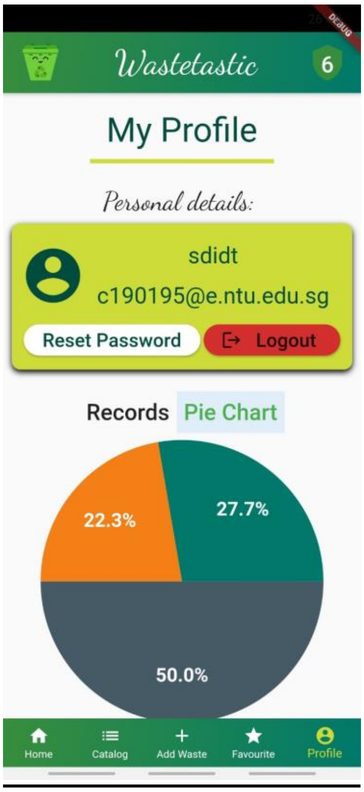
3.1.7. Profile Screen:



sdidt
c190195@e.ntu.edu.sg

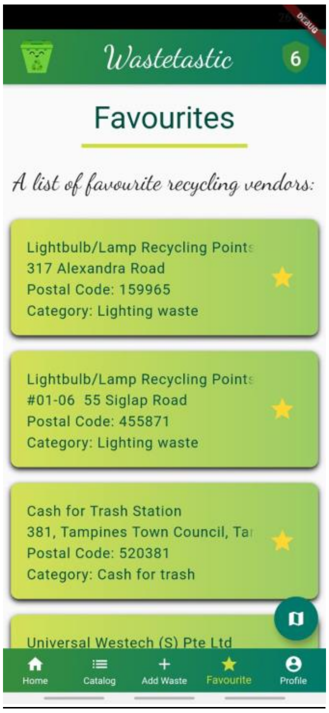
Cash for trash
2021-03-24 - 20:00:00
Weight:
8.7

E waste
2021-03-30 - 16:05:00
Weight:
7.8



sdidt
c190195@e.ntu.edu.sg

3.1.8. Favourites Screen:



A list of favourite recycling vendors:

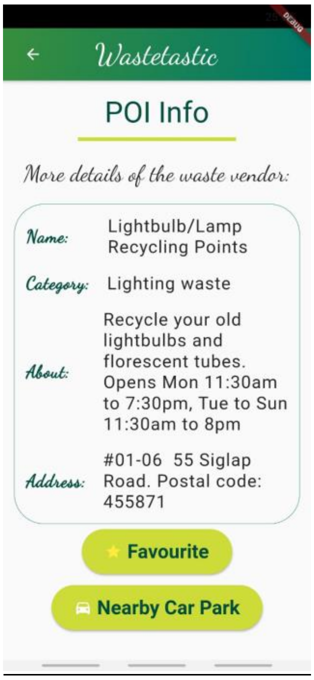
Lightbulb/Lamp Recycling Points
317 Alexandra Road
Postal Code: 159965
Category: Lighting waste

Lightbulb/Lamp Recycling Points
#01-06 55 Siglap Road
Postal Code: 455871
Category: Lighting waste

Cash for Trash Station
381, Tampines Town Council, Ta
Postal Code: 520381
Category: Cash for trash

Universal Westech (S) Pte Ltd

3.1.9. POI Details Screen:



POI Info

More details of the waste vendor:

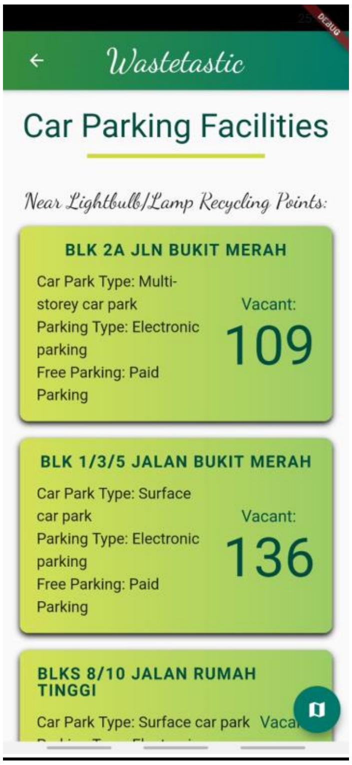
Name: Lightbulb/Lamp Recycling Points

Category: Lighting waste

About: Recycle your old lightbulbs and florescent tubes. Opens Mon 11:30am to 7:30pm, Tue to Sun 11:30am to 8pm

Address: #01-06 55 Siglap Road. Postal code: 455871

3.1.10. Carpark Details Screen



Car Parking Facilities

Near Lightbulb/Lamp Recycling Points:

BLK 2A JLN BUKIT MERAH

Car Park Type: Multi-storey car park
Parking Type: Electronic parking
Free Parking: Paid Parking

Vacant: 109

BLK 1/3/5 JALAN BUKIT MERAH

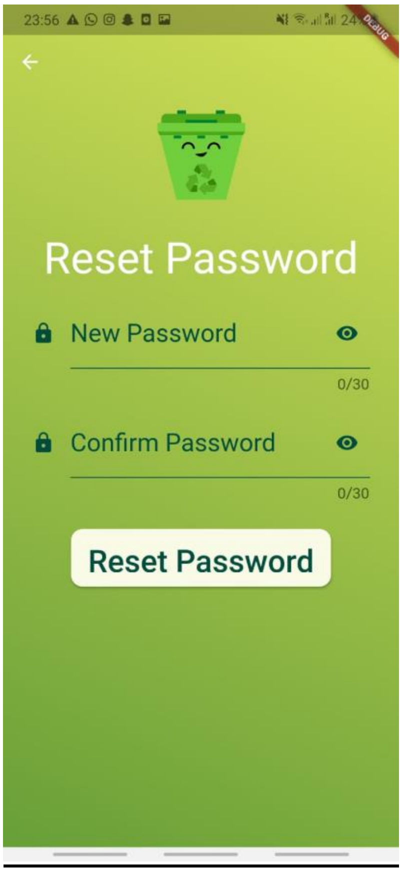
Car Park Type: Surface car park
Parking Type: Electronic parking
Free Parking: Paid Parking

Vacant: 136

BLKS 8/10 JALAN RUMAH TINGGI

Car Park Type: Surface car park Vaca

3.1.11. Reset Password Screen:



Reset Password

New Password

0/30

Confirm Password

0/30

Reset Password

3.2.Initial UI Mockup:

3.2.1. Login and Registration Screens:

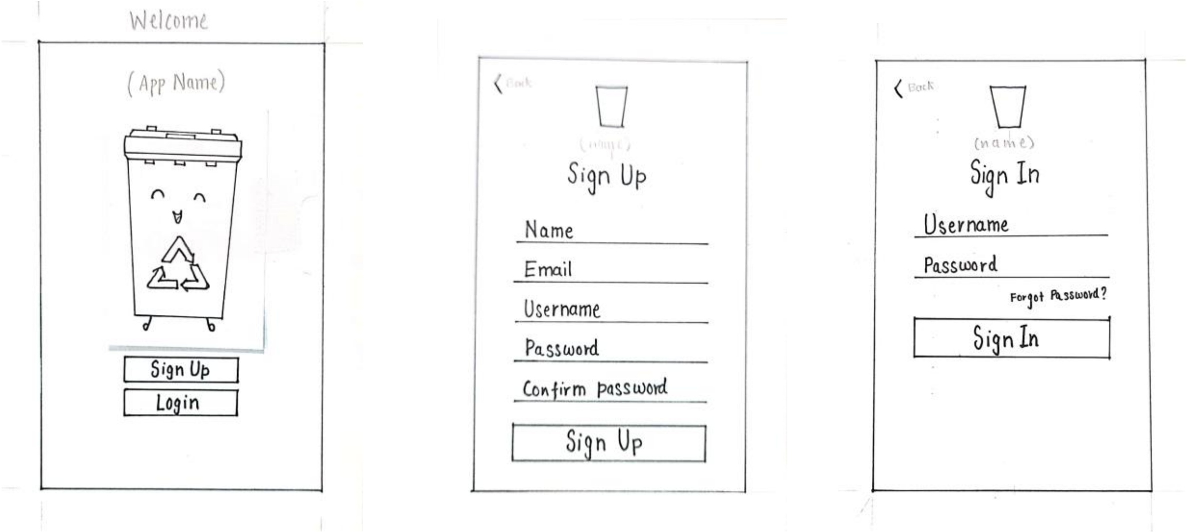


Figure 1: Login and Registration Screens

The above figure depicts the login and registration screens. The layout of these screens are simple and facilitate easy reading. Further, the screens are not flooded with information and give space for the eyes to rest.

3.2.2. Forgot Password and Reset Password Screen:

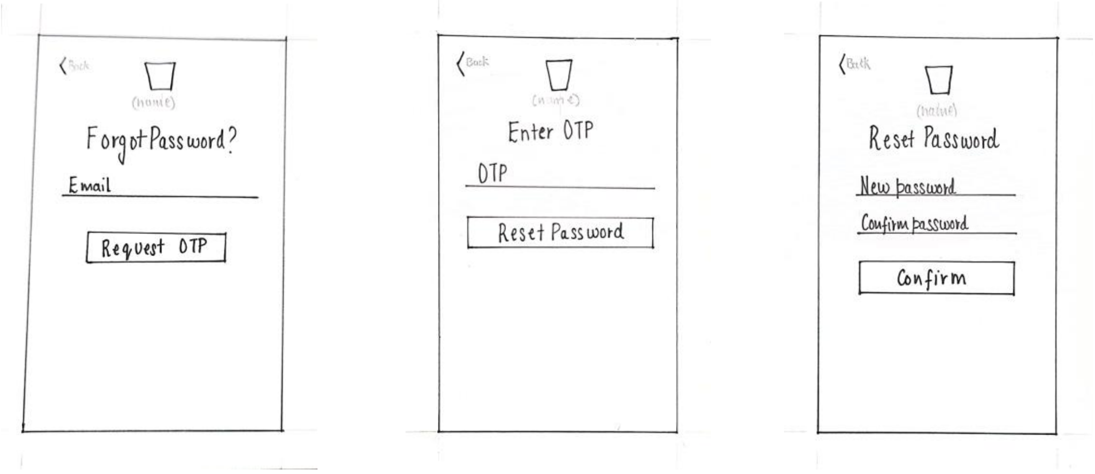


Figure 2: Forgot Password Screens

The above figure depicts the screens when the user selects the “Forgot Password?” option. The screens are very intuitive and simple to use. They also permit easy reversal of actions.

3.2.3. Home Screen

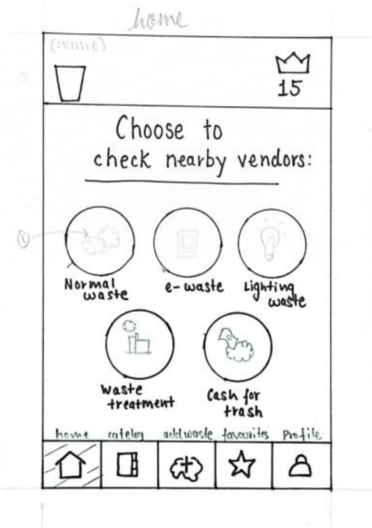


Figure 3: Home Screen

The above figure depicts the home screen after the suer has logged in. There are 5 options for the user to choose, which are all accompanied by a descriptive text. Further, the points accumulated by the user are displayed in the top right to incentivize the user to use the app more, and in turn, dispose waste more responsibly.

3.2.4. Screens for Searching Nearby Waste POIs:

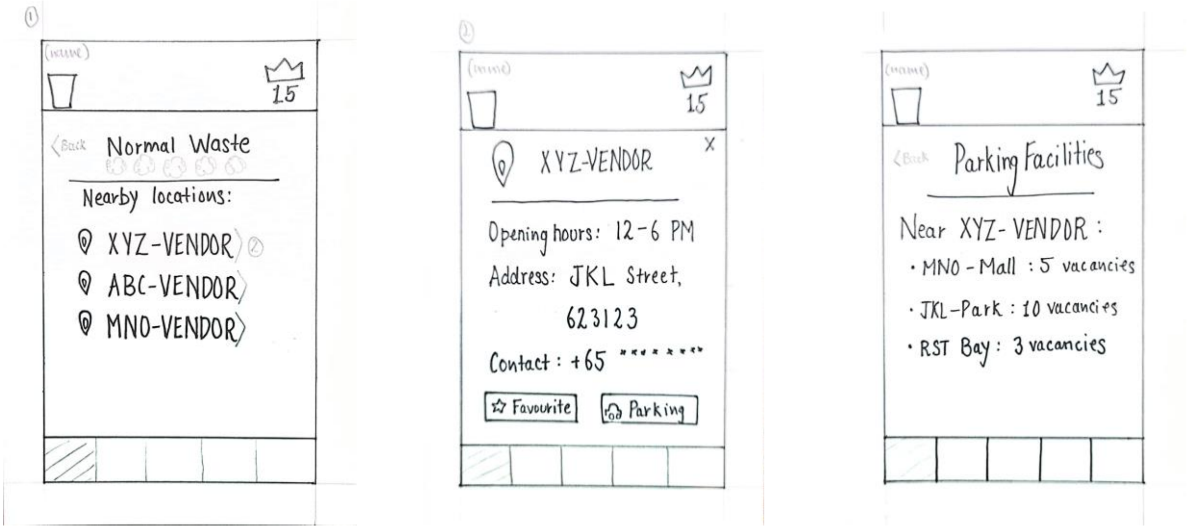


Figure 4: Search nearby waste POI screens

The above screens show the results when the user selects any option from the home screen, and chooses to view more specific details, as well as car parking availability.

3.2.5. General info on recycling screen:

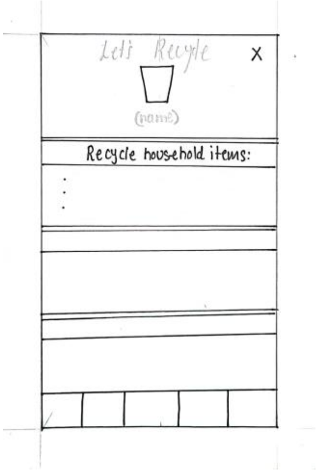


Figure 5: General info on recycling screen

The above figure depicts the screen which displays general information on recycling. It will guide the user on what items to recycle and has a very comfortable F-shaped reading pattern.

3.2.6. Catalog Tab Screen:

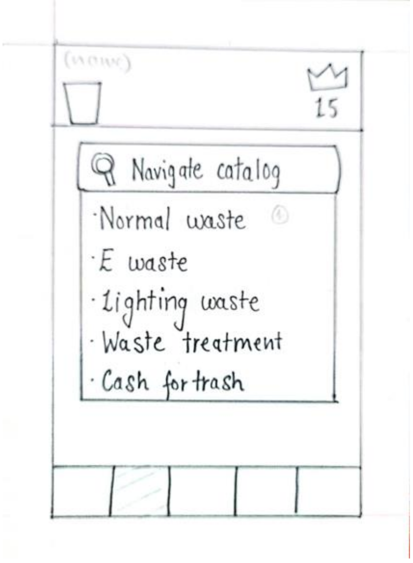


Figure 6: Catalog Tab Screens

The above figure depicts the catalog tab. The user has the option to choose any of the various filters, and the corresponding catalog will be displayed. Once again, an F-shaped reading pattern is employed to facilitate easy reading.

3.2.7. Add New Waste Record:

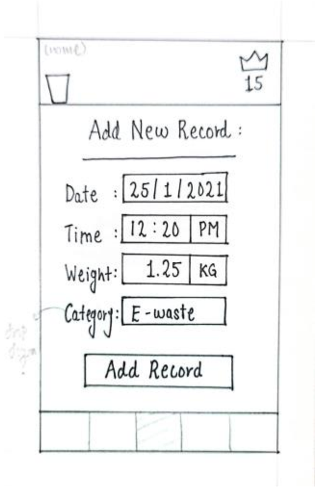


Figure 7: Add new waste record screen

The above screen depicts the “Add Waste Record” screen. The UI is designed in such a manner that errors can be prevented by forcing the user to choose from a drop-down menu for the category, and the date from a calendar.

3.2.8. Favorites Screen:

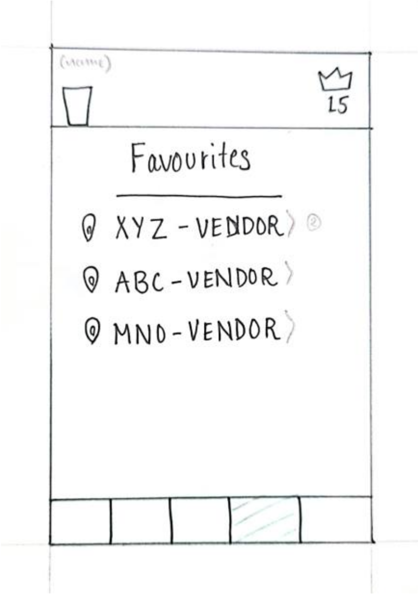


Figure 8: Favorites Screen

The above screen depicts the favorites screen. The user can read the list of his/her favorites, and tap on each one to view further details, as depicted in the results of the search in Figure 4.

3.2.9. Profile Screen:

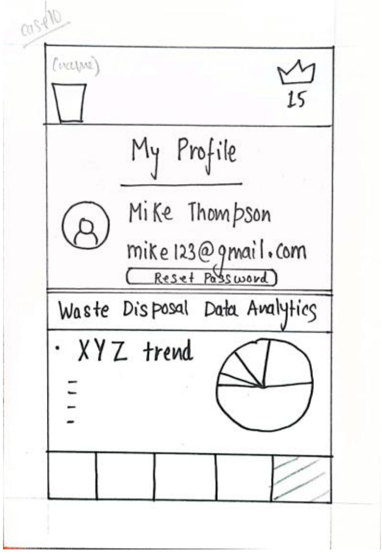


Figure 9: Profile Screen

The above figure depicts the profile screen. The user can view his personal information (except for password) as well as his personal waste disposal data analytics in a visual form, thus offering informative feedback.

3.3.Hardware Interfaces

The application is designed to be responsive and functional on most screen sizes of all Android and iOS mobile devices.

3.4.Software Interfaces

The application has been tested on different Android and iOS devices. The database used for this application is Firebase. We use the Email Password Sign In authentication.

3.5.Communications Interfaces

The registration and forgot password screen use the email_auth package to send the OTP to the user email.

Functional Requirements

4.1.USER REGISTRATION

4.1.1 The system must allow the user to create an account.

4.1.1.1 The system must allow the user to fill in his/her full name, email address, username, password and confirmation password.

4.1.1.1.1 The system must validate that all the text fields are filled up.

4.1.1.1.2 The system must validate that the input of the password text field contains at least 6 characters.

4.1.1.1.3 The system must validate that the password matches the confirmation password.

4.1.1.1.4 The system must validate that the email address entered is of appropriate format.

4.1.1.1.4.1 The email address must contain the "@" symbol.

4.1.1.1.4.2 The email address must contain a domain after the "@" symbol, containing at least one "." symbol.

4.1.1.1.5 The system must validate that the email address entered exists.

4.1.1.1.5.1 The system must send an OTP to the entered email address.

4.1.1.1.5.2 The system must prompt the user to enter the OTP.

4.1.1.1.5.3 The system must validate that the OTP entered matches the one sent to the entered email.

4.1.1.1.5.3.1 The system must give the user three tries to enter the OTP.

4.1.1.1.5.3.2 The system must redirect to the initial page after the user has made three incorrect attempts.

4.1.1.1.6 The system must validate that the email address has not already been registered.

4.1.1.1.7 The system must validate that the username has not already been registered.

4.1.1.1.8 The system must display the corresponding error message when any of the above-mentioned validation fails.

4.1.1.2 The system must redirect to the home page upon successful user registration.

4.2.USER LOGIN

4.2.1. The system must allow the user to login into the home page.

4.2.2. The system must be able to validate the username and password text fields.

4.2.2.1. The system must be able to validate that the username text field is filled.

4.2.2.2. The system must be able to validate that the password text field is filled.

4.2.2.3. The system must be able to validate that the username entered exists in the database.

4.2.2.4. The system must be able to validate that the password entered is the valid password corresponding to the username.

4.2.2.5. The system must display the corresponding error message when any of the above-mentioned validation fails.

4.2.3. The system must display a “Forgot Password?” option for the user at the login screen.

4.2.3.1. The system must prompt the user to enter the registered email address.

4.2.3.2. The system must validate that the email address is of appropriate format.

4.2.3.2.1. The email address must contain the “@” symbol.

4.2.3.2.2. The email address must contain a domain after the “@” symbol, containing at least one “.” symbol.

4.2.3.3. The system must send an OTP to the entered email address.

4.2.3.4. The system must prompt the user to enter the OTP sent.

4.2.3.4.1. The system must validate that the OTP entered by the user matches the OTP sent to the email address.

4.2.3.4.1.1. The system must give the user three tries to enter the OTP.

4.2.3.4.1.2. The system must redirect to the initial page after the user has made three incorrect attempts.

4.2.3.5. The system must prompt the user for the new password and confirm the new password.

4.2.3.5.1. The system must validate that the entered password matches the confirmation password.

4.2.4. The system must redirect the user to the home page upon successful login.

4.3. USER

4.3.1. The user domain must have the following 5 components- Home Screen, Catalog Tab, Add Waste Record, Favourites and Profile.

4.3.2. The Home Screen must display the progress of the user in the disposal of trash.

4.3.2.1. The system must display the total points earned by the user until now.

4.3.2.2. The system must allow the user to select the type of waste he/she would like to recycle or dispose at that point.

4.3.2.3. The system must display all the waste POI locations near the current location of the user.

4.3.2.4. The system must on selection of a particular waste POI, display to the user the information of that waste POI.

4.3.3. The system must allow the user to view the catalog of categories of waste disposed and recycled in Singapore in the Catalog tab.

4.3.3.1. The system must have a waste category selected by default.

4.3.3.2. The system must allow the user to choose a category from the list provided according to his/her wish.

4.3.3.3. The system must display a list of all the different locations of waste POIs under the category chosen.

4.3.3.3.1. The system must allow the user to choose and view a specific location from the list.

4.3.3.3.1.1. The system must display more information on the specific location chosen by the user.

4.3.3.3.1.2. The system must allow the user to add the chosen location to the favourites section.

4.3.4. The system must allow the user to enter its personal amount of waste disposed or recycled in the Add Waste Record page.

4.3.4.1. The system must allow the user to select the date and time when he/she disposed of or recycled the waste.

4.3.4.1.1. The system must validate that the user has selected a date.

4.3.4.1.2. The system must validate that the user has entered a time.

4.3.4.1.3. The system must validate that the date chosen by the user is valid.

4.3.4.1.4. The system must validate that the time entered by the user is valid.

4.3.4.1.5. The system must display the corresponding error message when any of the above-mentioned validation fails.

4.3.4.2. The system must allow the user to enter the weight and type of waste disposed or recycled.

4.3.4.2.1. The system must validate the numeric field is filled.

4.3.4.2.2. The system must validate the text field is filled.

4.3.4.2.3. The system must validate that the weight entered is valid.

4.3.4.2.4. The system must validate that the category entered is valid.

4.3.4.2.5. The system must display the corresponding error message when any of the above-mentioned validation fails.

4.3.4.3. The system must display a suitable message to the user upon successful storage of his/her waste records into the database.

4.3.5. The user must be able to view the locations of waste POI that he/she starred as a favourite in the Favourites page.

4.3.5.1. The system must allow the user to choose a specific location if there is more than one location under Favourites.

4.3.5.1.1. The system must display information on the location chosen by the user.

4.3.6. The user account details, waste record history, points earned and reset password must be displayed in the Profile page.

4.3.6.1. The system must display the personal account details of the user except for the password.

4.3.6.2. The system must display the points earned by the user until now.

4.3.6.3. The system must provide visual analytics of the amount of weight disposed or recycled by the user.

4.3.6.3.1. The system must display a pie chart to the user showing the percentage of waste disposed or recycled in different categories.

4.3.6.4. The system must display the amount of waste disposed or recycled by the user, category-wise, in kilograms.

4.3.6.5. The system must display the total amount of waste disposed or recycled by the user in kilograms.

4.3.6.6. The system must allow the user to change his/her old password to new password using the Reset Password option.

4.3.6.6.1. The system must prompt the user for the old password.

4.3.6.6.1.1. The system must validate the text field is filled.

4.3.6.6.1.2. The system must validate that the password entered matches with the one in the database currently.

4.3.6.6.1.3. The system must display the corresponding error message when any of the above-mentioned validation fails.

4.3.6.6.2. The system must prompt the user to enter the new password and confirm the new password.

4.3.6.6.2.1. The system must validate the text fields are filled.

4.3.6.6.2.2. The system must validate that the input of the new password text field contains at least 6 characters, including at least one special character and at least one digit.

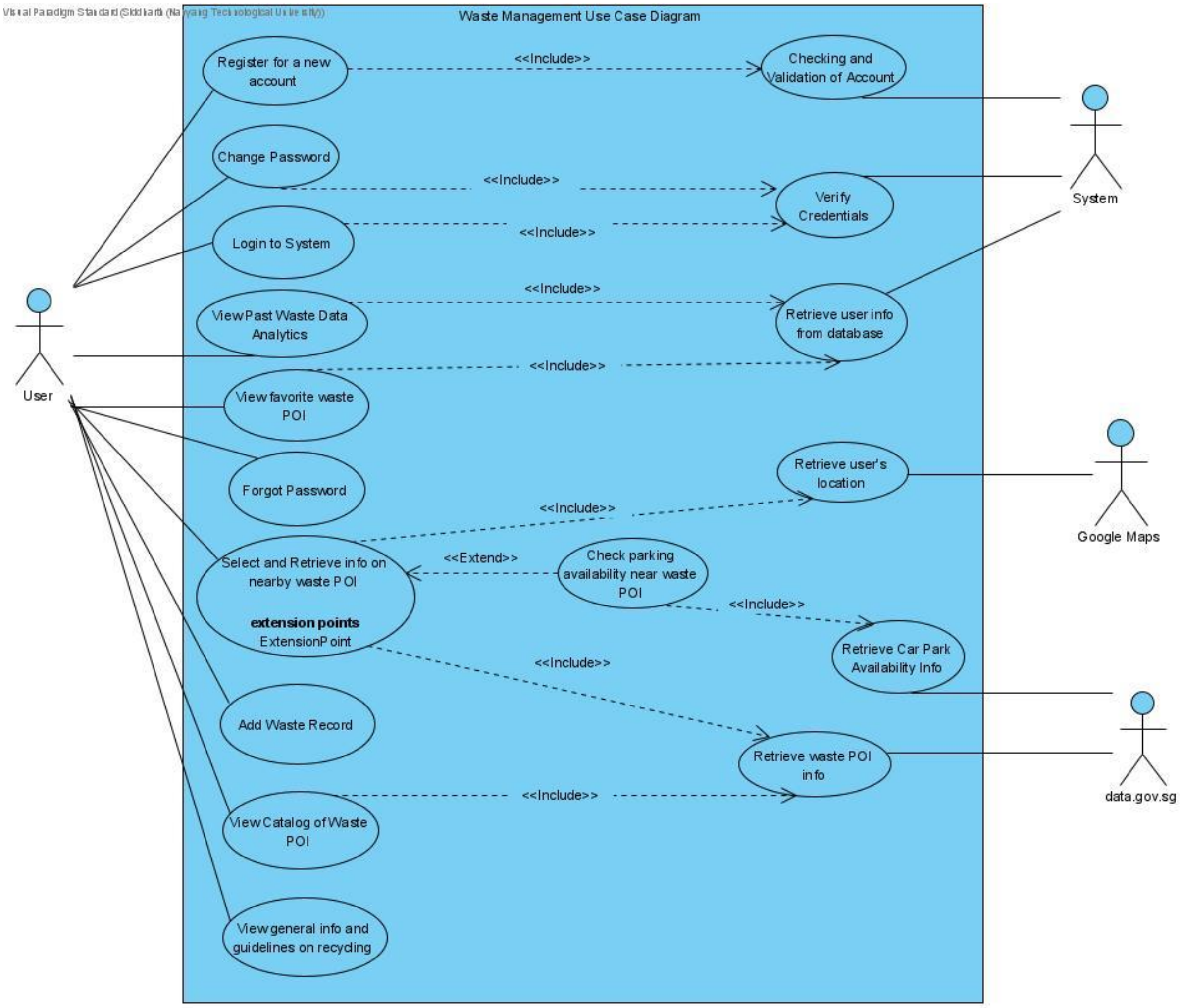
4.3.6.6.2.3. The system must validate that the new password matches the new password confirmation.

4.3.6.6.2.4. The system must display the corresponding error message when any of the above-mentioned validation fails.

4.3.6.6.3. The system must send an email to the user's registered email address on successful resetting of the password.

4.3.7. The system must allow the user to view general guidelines and information on recycling by tapping on the app icon visible on the screen.

4.4.USE CASE DIAGRAM


Visual Paradigm Standard (Siddharth (Nanyang Technological University))
Waste Management Use Case Diagram
Register for a new account
<<Include>>
Checking and Validation of Account
Change Password
<<Include>>
Verify Credentials
Login to System
<<Include>>
<<Include>>
View Past Waste Data Analytics
Retrieve user info from database
<<Include>>
View favorite waste POI
Forgot Password
Retrieve user's location
<<Include>>
Select and Retrieve info on nearby waste POI
extension points
ExtensionPoint
<<Extend>>
Check parking availability near waste POI
<<Include>>
Retrieve Car Park Availability Info
<<Include>>
Add Waste Record
Retrieve waste POI info
View Catalog of Waste POI
<<Include>>
View general info and guidelines on recycling
User
System
Google Maps
data.gov.sg

4.5.USE CASE DESCRIPTIONS

Use Case ID:	1		
Use Case Name:	Register for a new account		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	User
Description:	Registration for a new user account
Preconditions:	1. Username and Email Address are not present in the database. 2. User is connected to the internet (Wi-Fi/Mobile Data)
Postconditions:	1. New user account is present in the database. 2. User receives a confirmation email for registration.
Priority:	High
Frequency of Use:	1-3 times per lifetime
Flow of Events:	1. User fills ‘Full Name’, ‘Email Address’, ‘Username’, ‘Password’ and ‘Confirm Password’ fields. 2. User clicks the ‘Sign Up’ button.

	3. System checks if ‘Password’ and ‘Confirm Password’ are identical. 4. System uses the included use case ‘Checking and Validation of Account’ to validate and confirm registration. 5. System redirects the user to the home screen upon successful registration.
Alternative Flows:	AF-2: The user does not fill up every field. 1. System prompts the user to fill in all the missing fields. 2. User enters the missing details. 3. Return to Step 2. AF-2: The password is less than 6 characters long. 1. System prompts the user to enter a password at least 6 characters in length. 2. User re-enters a valid password 3. Return to Step 2. AF-3: ‘Password’ and ‘Confirm Password’ are not identical 1. System display error message, “The passwords do not match. Please re-enter your password.” 2. User re-enters ‘Password’ and ‘Confirm Password’ fields correctly. 3. Return to Step 2.
Exceptions:	-
Includes:	1. Checking and validation of Account
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	2		
Use Case Name:	Checking and Validation of Account		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	System
Description:	Validate account availability, send OTP to email address and send confirmation email to registered email address,
Preconditions:	1. User is connected to the internet (Wi-Fi/Mobile Data)
Postconditions:	1. New user account is present in the database. 2. User receives a confirmation email for registration.
Priority:	High
Frequency of Use:	1-3 times per lifetime

Flow of Events:	1. System will validate that the username is not present in the database. 2. System will validate that the email address is not present in the database. 3. System will send OTP to the email address. 4. User enters the OTP when prompted. 5. System verifies the entered OTP. 6. System will register the account and send a confirmation email.
Alternative Flows:	AF-1: Username already exists in database 1. System display error message, “Username already registered. Please choose a different username.” 2. User re-enters their username and clicks on the ‘Sign up’ button. 3. Return to Step 1. AF-2: Email Address already exists in database 1. System display error message, “Email Address already registered. Please login using the email address or enter a different email address.” 2. User re-enters their email address and clicks on the ‘Register’ button. 3. Return to Step 1. AF-5: User does not enter/enters wrong OTP 1. System displays an error message, “OTP entered is incorrect/left blank” and prompts the user to enter the OTP again. 2. Return to step 4.
Exceptions:	EX-AF-5: User enters the wrong OTP 3 times. 1. System displays an error message, “You have exceeded the maximum limit of tries for OTP. Please restart the registration process.” 2. System redirects the user to the signup page.
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	3		
Use Case Name:	Change Password		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	User
Description:	Allow the User to change account password
Preconditions:	1. User is logged into his account. 2. User is connected to the internet (Wi-Fi/Mobile Data)
Postconditions:	1. User receives a confirmation email of password change. 2. User account password is changed to the new password.

Priority:	Medium
Frequency of Use:	6-8 times per lifetime
Flow of Events:	1. User fills ‘Username’, ‘Current Password’, ‘New Password’ and ‘ConfirmNew Password’ fields. 2. User clicks the ‘Change Password’ button. 3. System checks if ‘New Password’ and ‘Confirm New Password’ are identical. 4. System uses the included use case ‘Verify Credentials’ to validate the registered Username and Current Password. 5. System updates the user Password in the database successfully.
Alternative Flows:	AF-3: ‘New Password’ and ‘ConfirmNew Password’ are not identical 1. System displays an error message, “The passwords do not match. Please re-enter your password.” 2. User re-enters the “New Password” and “Confirm New Password” fields. 3. Return to Step 2. AF-4: User enters wrong ‘Username’ and/or ‘Current Password’ 1. System displays an error message, “Incorrect Username or Password”. 2. User re-enters the “Username” and “Current Password” fields. 3. Return to Step 2.
Exceptions:	-
Includes:	1. Verify Credentials
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	4		
Use Case Name:	Verify Credentials		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	System
Description:	Verify user authentication using username and password
Preconditions:	1. User account is present in the database. 2. User is connected to the internet (Wi-Fi/Mobile Data)
Postconditions:	1. The User must be able to access their account.
Priority:	Medium

Frequency of Use:	1-3 times daily
Flow of Events:	1. System verifies the user login credentials with the user database. 2. System successfully authenticates the User.
Alternative Flows:	AF-1: Entered credentials are incorrect. 1. System displays an error message, “Incorrect Username or Password”. 2. User re-enters login credentials. 3. Return to Step 1.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	5		
Use Case Name:	Login to System		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	User
Description:	Verify user authentication using username and password
Preconditions:	1. User account is present in the database. 2. User is connected to the internet (Wi-Fi/Mobile Data)
Postconditions:	1. The User must be able to access their account.
Priority:	Medium

Frequency of Use:	1-3 times daily
Flow of Events:	1. User enters ‘Username’ and ‘Password’. 2. User clicks the ‘Login’ button. 3. System uses the included use case ‘Verify Credentials’ to validate user login credentials.
Alternative Flows:	-
Exceptions:	-
Includes:	1. Verify Credentials
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	6		
Use Case Name:	Forgot Password		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	User
Description:	Allow user to reset password in case he/she has forgotten password
Preconditions:	1. User account is present in the database. 2. User is connected to the internet (Wi-Fi/Mobile Data)

Postconditions:	1. The User must be able to access their account. 2. The User must receive a confirmation email of password change.
Priority:	Medium
Frequency of Use:	1-3 times per lifetime
Flow of Events:	1. User enters the registered email. 2. System sends password reset link to the corresponding email address. 3. User enters the “New Password” at the link provided. 4. System updates the user password and redirects the user to the home screen.
Alternative Flows:	AF-2: The entered email does not exist in the database. 1. System displays an error message, “Entered email does not exist in the database.” and prompts the user to enter the email again. 2. Return to Step 1.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	7		
Use Case Name:	Select and Retrieve info on nearby waste POI		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	User
Description:	To view info on nearby waste POI
Preconditions:	1. User is logged into the account. 2. User is connected to the internet (Wi-Fi/Mobile Data) 3. User must have location access enabled for the app
Postconditions:	1. Location and details of a nearby waste POIs are displayed to the user.
Priority:	High
Frequency of Use:	1-3 times daily
Flow of Events:	1. User navigates to the home screen. 2. User chooses the kind of waste POIs they would like to view info on. 3. System uses the included use case “Retrieve user’s location”. 4. System uses the included use case “Retrieve waste POI info” and display waste POIs near the user's location. 5. User then chooses a specific waste POI from the results. 6. System displays info on selected waste POI. 7. User may choose to view nearby car parking facilities and/or mark the waste POI as favourite.
Alternative Flows:	-
Exceptions:	-
Includes:	1. Retrieve waste POI info 2. Retrieve user’s location
Special Requirements:	-
Assumptions:	-

Notes and Issues:	-

Use Case ID:	8		
Use Case Name:	Check parking availability near waste POI		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	User
Description:	To search for car parking availability near the waste POI
Preconditions:	1. User is logged in to the account. 2. User is connected to the internet (Wi-Fi/Mobile Data) 3. User must have location access enabled for the app
Postconditions:	1. Location and details of a nearby car park facility must be displayed.
Priority:	Medium
Frequency of Use:	1-3 times weekly
Flow of Events:	1. System uses the included use case “Retrieve car park availability info”. 2. System displays the car park availability near the Waste POI. 3. User can then choose a convenient car park.
Alternative Flows:	-
Exceptions:	-
Includes:	1. Retrieve car park availability info
Special Requirements:	-

Assumptions:	-
Notes and Issues:	-

Use Case ID:	9		
Use Case Name:	Add Waste Record		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	User
Description:	To keep track of user's personal waste disposal data
Preconditions:	1. User must be logged in to his/her account. 2. User is connected to the internet (Wi-Fi/Mobile Data)
Postconditions:	1. User's personal waste disposal info must be saved to the database.
Priority:	High
Frequency of Use:	0-10 times per day
Flow of Events:	1. User navigates to the "Add Waste Record" page. 2. User selects the type of waste from a drop-down menu. 3. User enters the rough date and time when the waste was disposed/recycled. 4. User keys in the weight of waste (in kgs). 5. User presses add record. 6. System displays a message confirming the addition of waste record to the user database.

Alternative Flows:	AF-4-1: System detects that the user has not filled in a required field - waste type, date, time, weight of waste. 1. Return to step 2. AF-4-2: System detects that the user has not keyed in any value for the weight. 1. Return to step 4. AF-4-3: User can optionally add a picture of the waste. 1. System verifies that the image does not exceed the maximum size threshold. 2. Return to step 4
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	1. User is trustworthy and will not willfully key in incorrect details.
Notes and Issues:	-

Use Case ID:	10		
Use Case Name:	View past waste data analytics		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	User
Description:	To display the user's past personal waste data analytics.
Preconditions:	1. User must be logged in to his/her account 2. User is connected to the internet (Wi-Fi/Mobile Data)
Postconditions:	1. User's waste disposal data analytics must be displayed.
Priority:	High
Frequency of Use:	0-10 times per day
Flow of Events:	1. User navigates to the Profile page. 2. System uses the included use case "Retrieve user info from database". 3. The system displays the retrieved data in the form of a pie chart along with other statistics. 4. User scrolls down the "Waste Disposal Data Analytics" section to view his/her past activity.
Alternative Flows:	-
Exceptions:	EX-4: User has not recorded any waste disposals 1. The system displays an error message, "No available waste disposal records."
Includes:	1. Retrieve user info from database.
Special Requirements:	-

Assumptions:	-
Notes and Issues:	-

Use Case ID:	11		
Use Case Name:	View Favorite Waste POI		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	User
Description:	To view the favorite waste POI
Preconditions:	1. User is logged in to the account. 2. User is connected to the internet (Wi-Fi/Mobile Data)
Postconditions:	1. User's favorite waste POIs must be displayed.
Priority:	Medium
Frequency of Use:	1-3 times weekly
Flow of Events:	1. User navigates to the “Favorites” page. 2. System uses the included use case “Retrieve user info from database”.

	3. System displays the list of user's favorites. 4. User can then choose a specific waste POI to view more info.
Alternative Flows:	-
Exceptions:	EX-3: User has not marked any favorites. 1. System displays an error message "You are yet to add any favorites."
Includes:	1. Retrieve user info from database
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	12		
Use Case Name:	View Catalog of Waste POI		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	User
Description:	To view the catalog of specific type of waste POI
Preconditions:	1. User is logged in to the account. 2. User is connected to the internet (Wi-Fi/Mobile Data)
Postconditions:	1. Catalog of specific type of waste POI must be displayed.

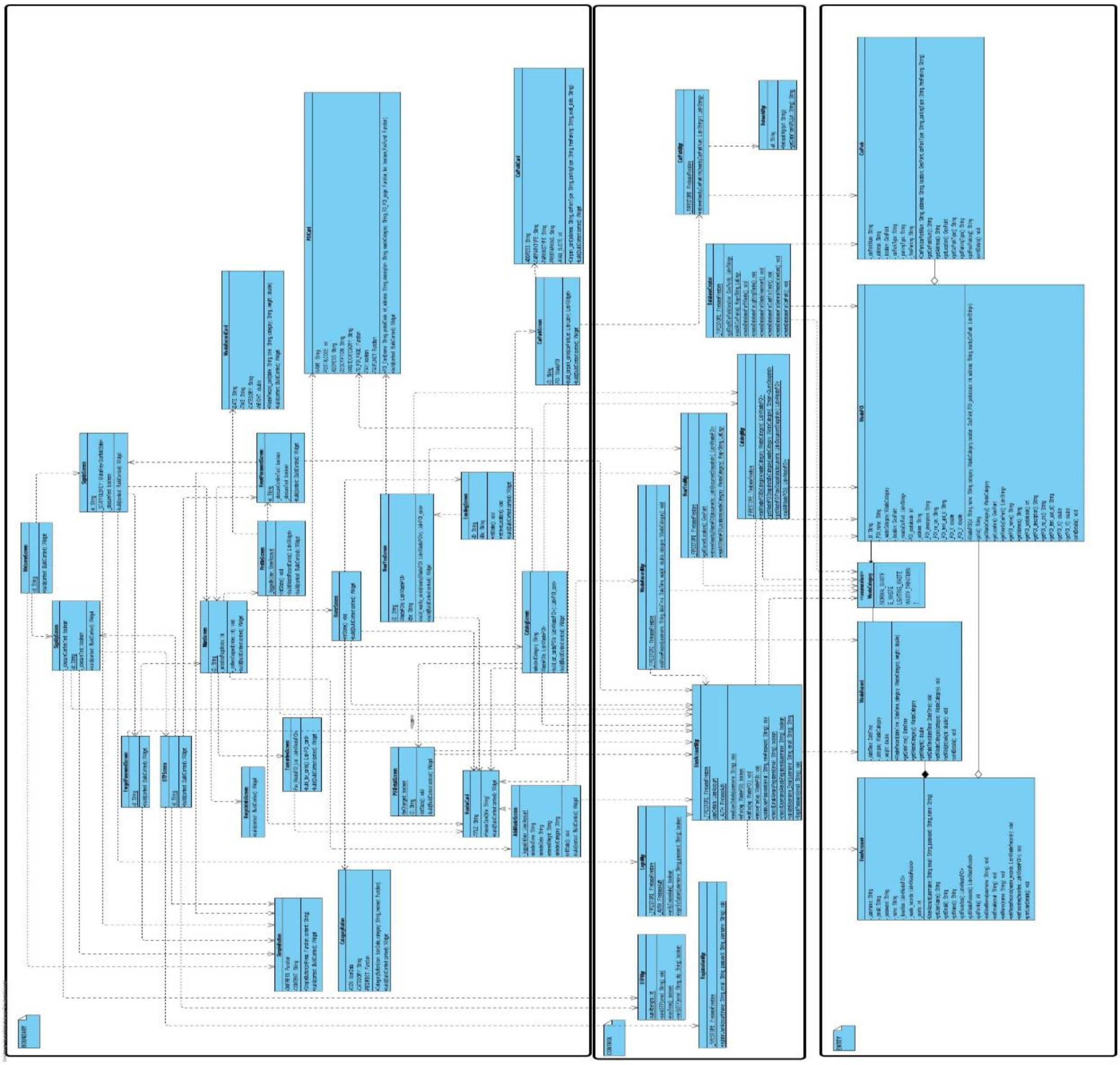
Priority:	Medium
Frequency of Use:	1-3 times daily
Flow of Events:	1. User navigates to the "Catalog" page. 2. System uses the included use case "Retrieve waste POI info". 3. System displays the list of waste POIs of the selected filter. 4. User can change the filter to view waste POIs of other categories.
Alternative Flows:	-
Exceptions:	-
Includes:	1. Retrieve waste POI info
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	13		
Use Case Name:	View General Info and guidelines of recycling		
Created By:	Agnesh, Siddharth	Last Updated By:	
Date Created:	09/02/2021	Date Last Updated:	

Actor:	User
Description:	To view the general info and guidelines on recycling

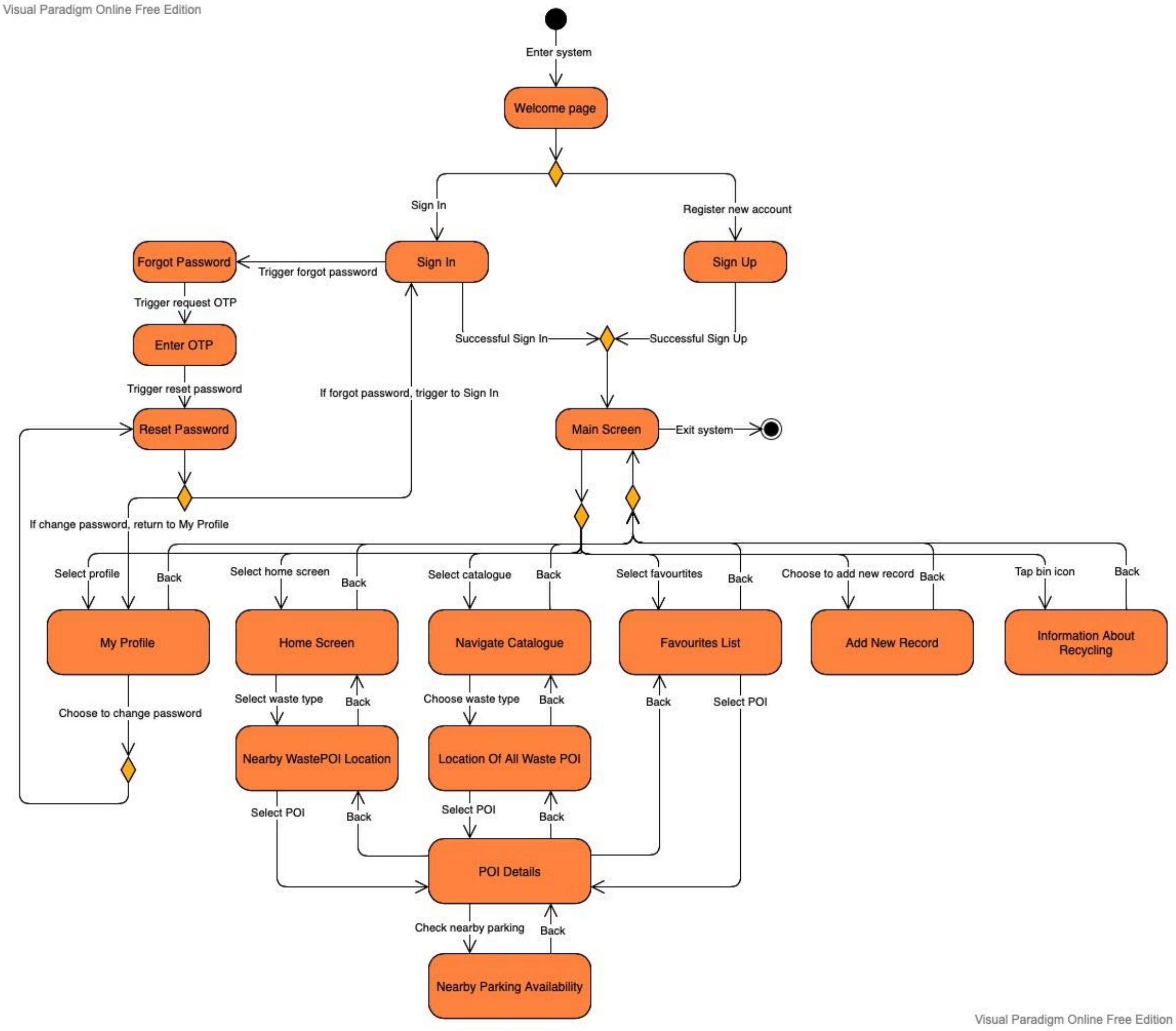
Preconditions:	1. User is logged in to the account. 2. User is connected to the internet (Wi-Fi/Mobile Data)
Postconditions:	1. User must be able to view general info and guidelines on recycling.
Priority:	Medium
Frequency of Use:	1-3 times weekly
Flow of Events:	1. User taps the app icon - a dustbin with a recycling symbol. 2. System displays general info and guidelines on recycling.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

4.6.CLASS DIAGRAM



4.7.DIALOG MAP

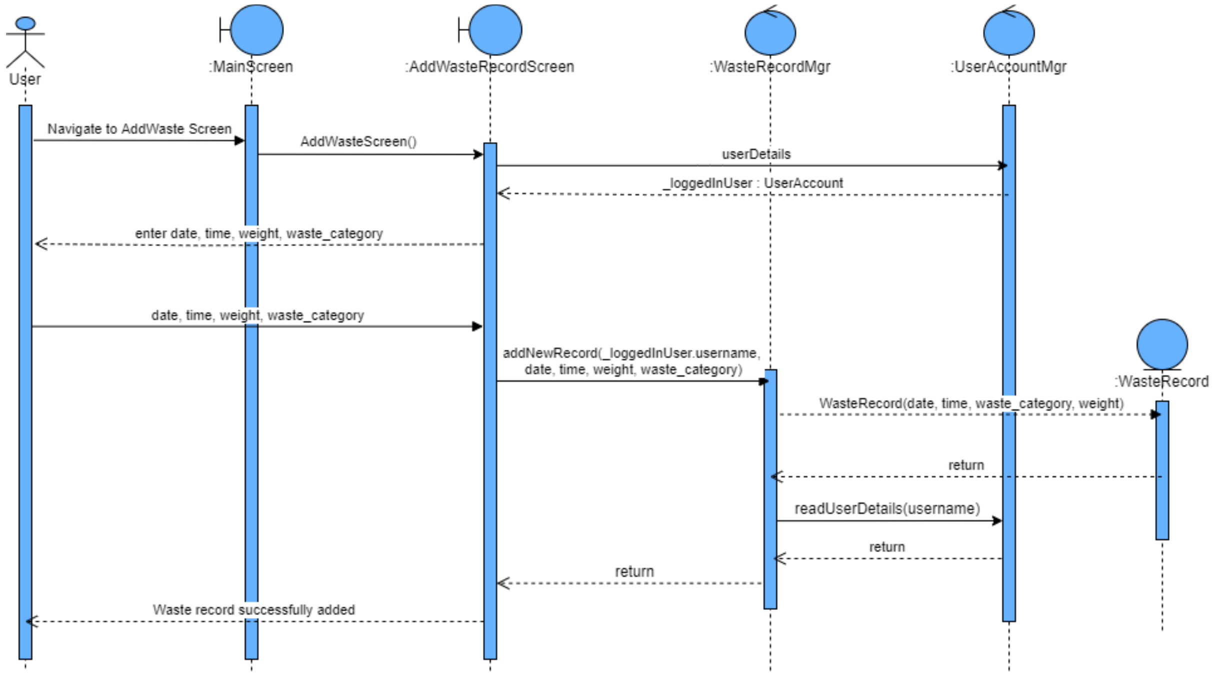
Visual Paradigm Online Free Edition



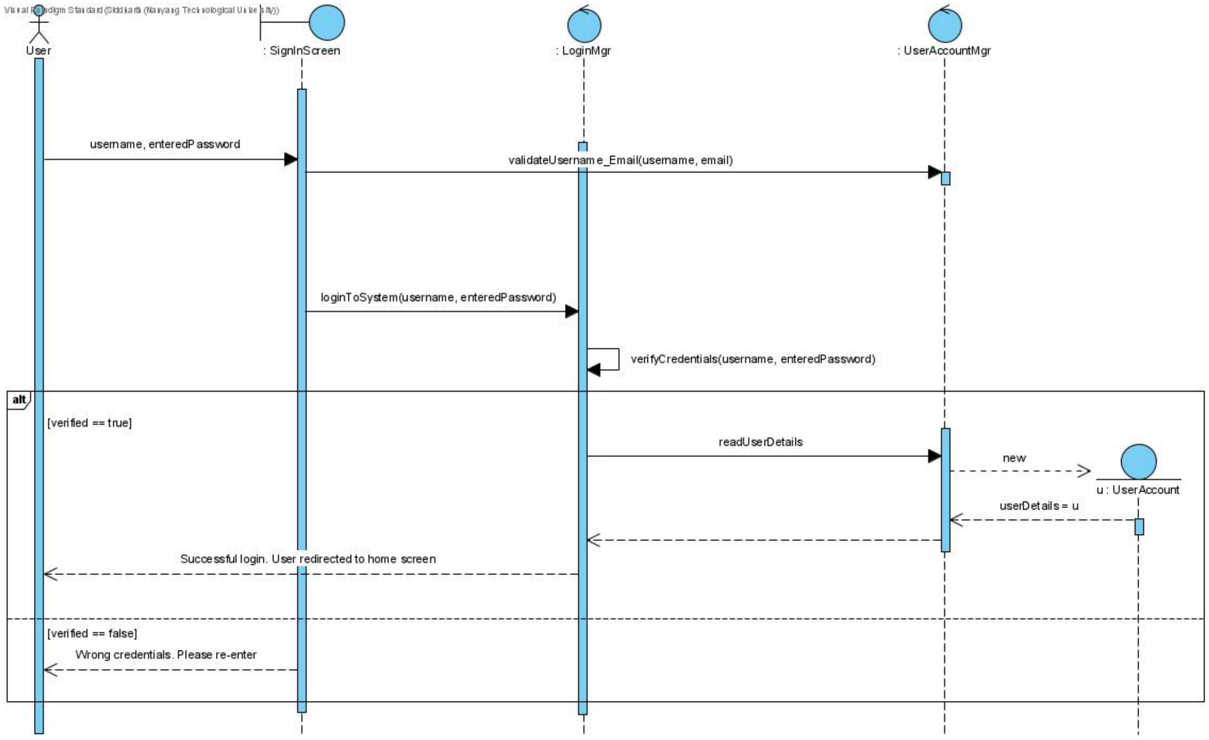
Visual Paradigm Online Free Edition

4.8.SEQUENCE DIAGRAMS

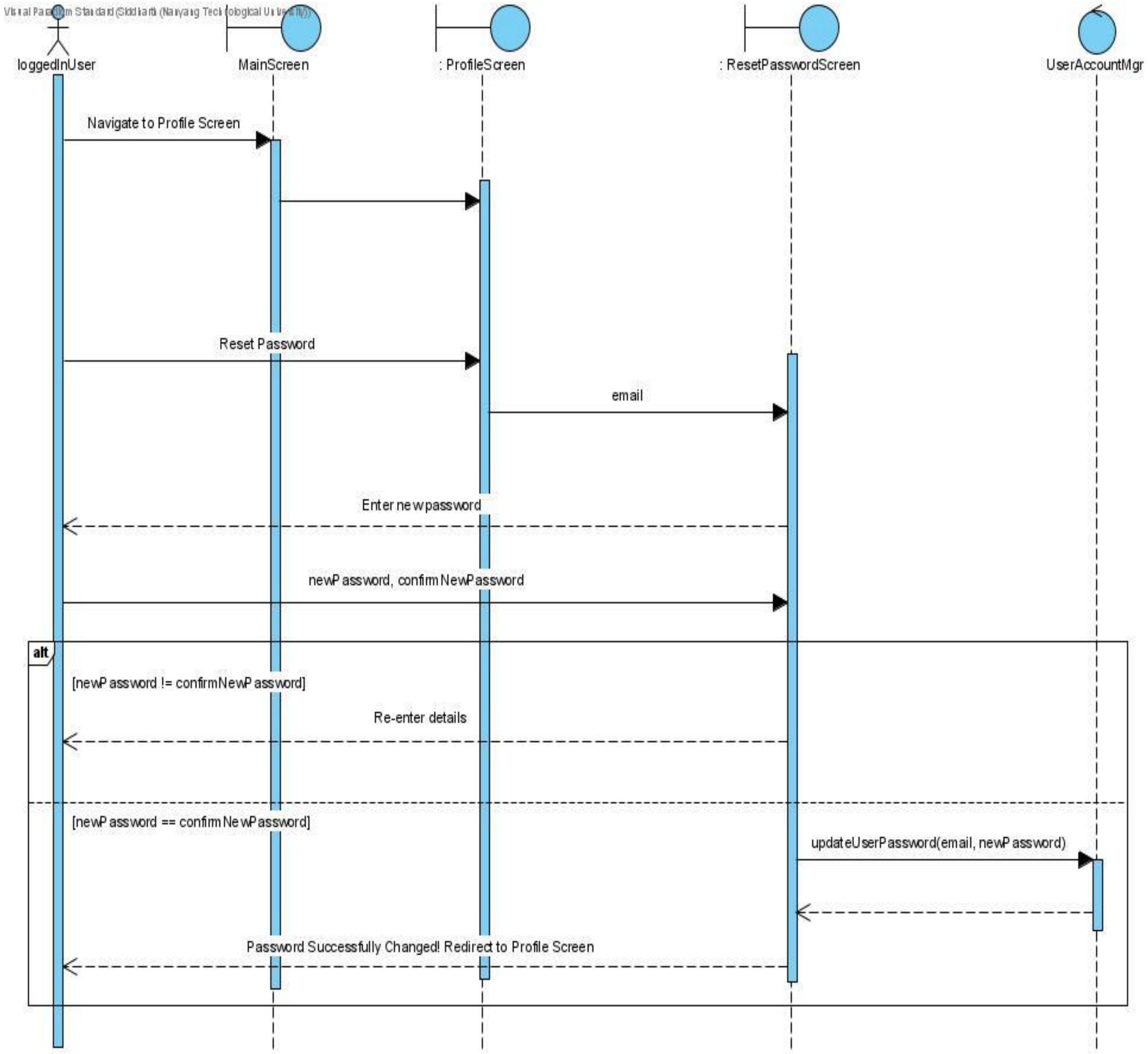
4.8.1. Add Waste Record


User
:MainScreen
:AddWasteRecordScreen
:WasteRecordMgr
:UserAccountMgr
Navigate to AddWaste Screen
AddWasteScreen()
userDetails
_loggedInUser : UserAccount
enter date, time, weight, waste_category
date, time, weight, waste_category
addNewRecord(_loggedInUser.username, date, time, weight, waste_category)
:WasteRecord
WasteRecord(date, time, waste_category, weight)
return
readUserDetails(username)
return
return
Waste record successfully added

4.8.2. Login

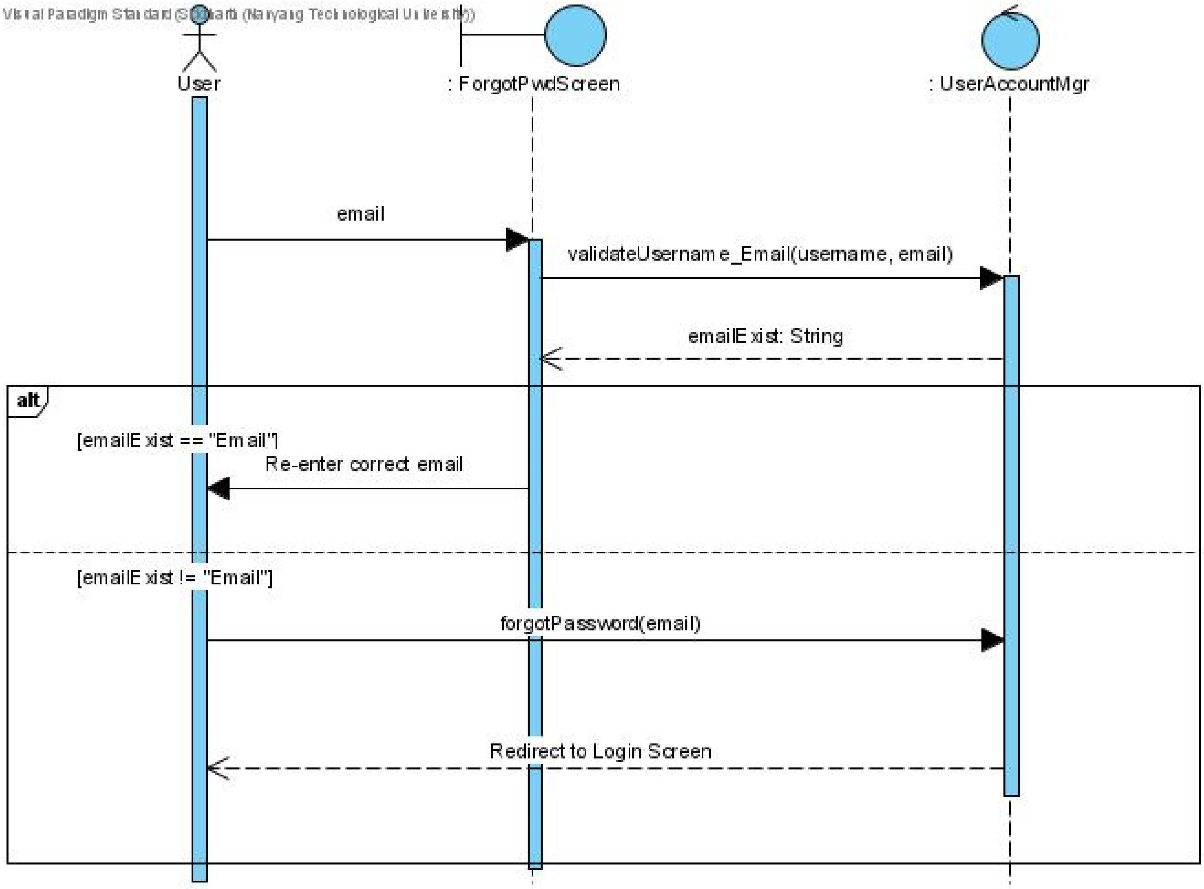

User
: SignInScreen
: LoginMgr
: UserAccountMgr
username, enteredPassword
validateUsername_Email(username, email)
loginToSystem(username, enteredPassword)
verifyCredentials(username, enteredPassword)
alt
[verified == true]
readUserDetails
new
u : UserAccount
userDetails = u
Successful login. User redirected to home screen
[verified == false]
Wrong credentials. Please re-enter

4.8.3. Change Password

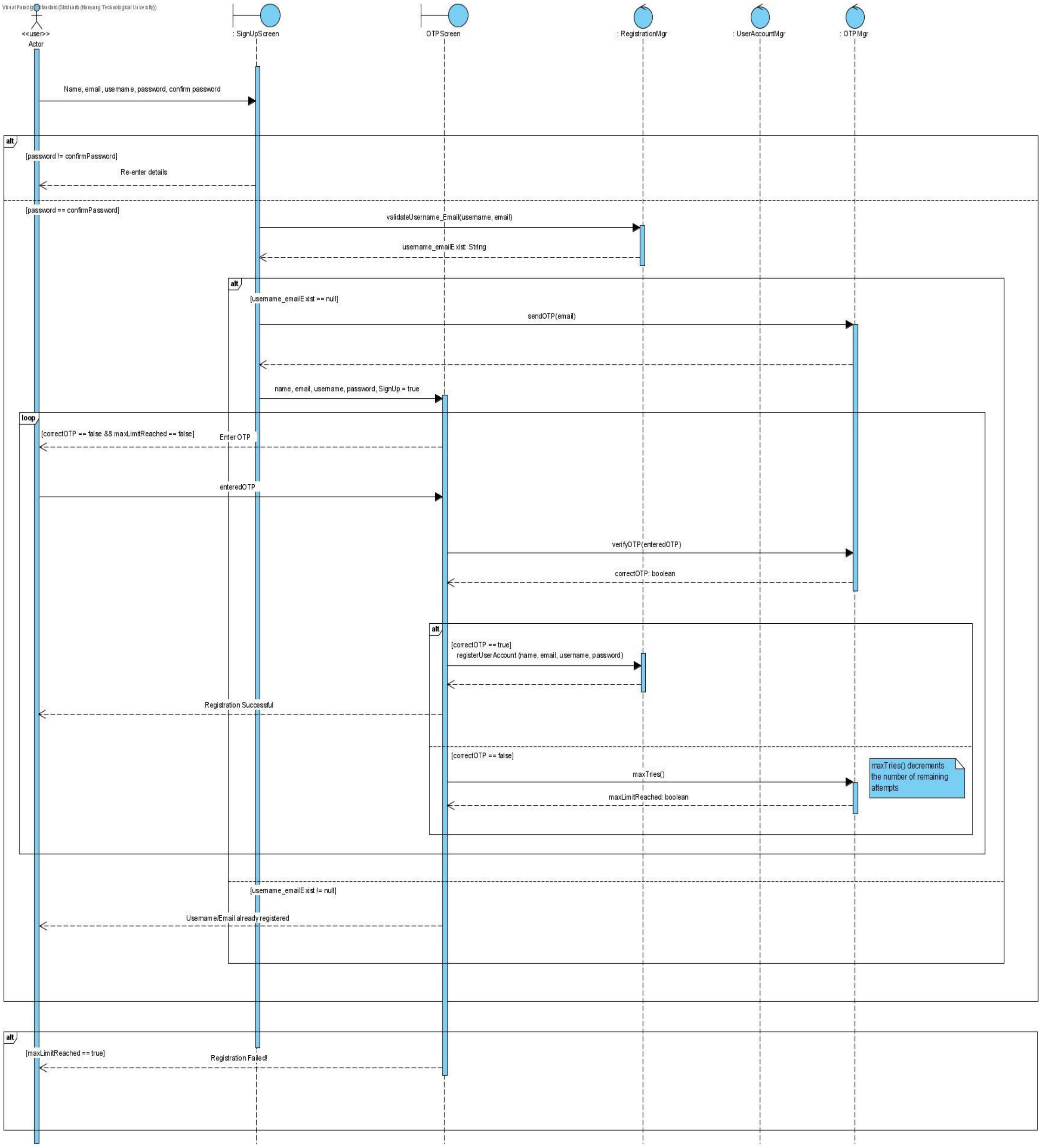

loggedInUser
MainScreen
: ProfileScreen
: ResetPasswordScreen
UserAccountMgr
Navigate to Profile Screen
Reset Password
email
Enter new password
newPassword, confirmNewPassword
alt
[newPassword != confirmNewPassword]
Re-enter details
[newPassword == confirmNewPassword]
updateUserPassword(email, newPassword)
Password Successfully Changed! Redirect to Profile Screen

Visual Paradigm Standard(Siddharth (Nanyang Technological University))

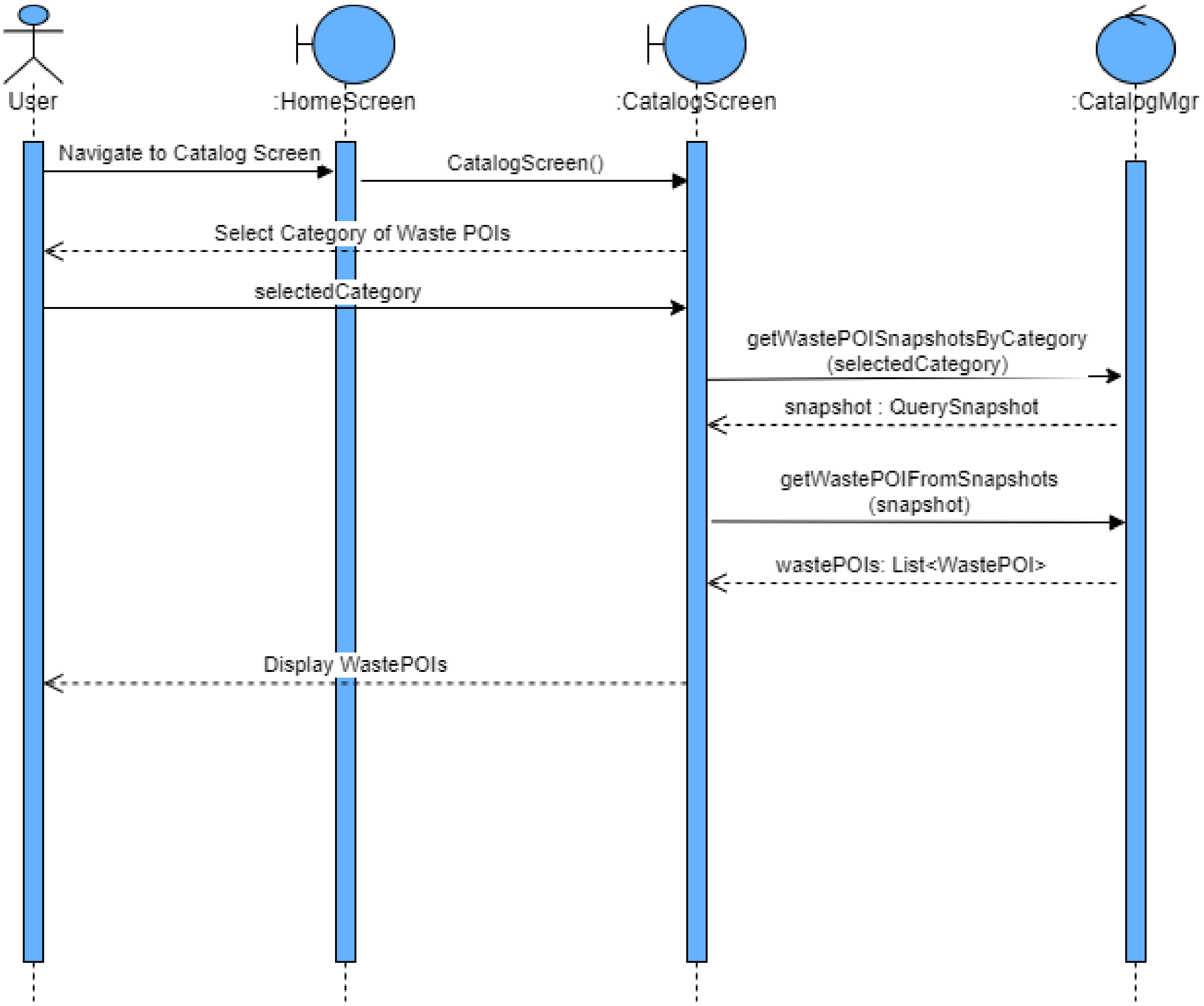
4.8.4. Forgot Password



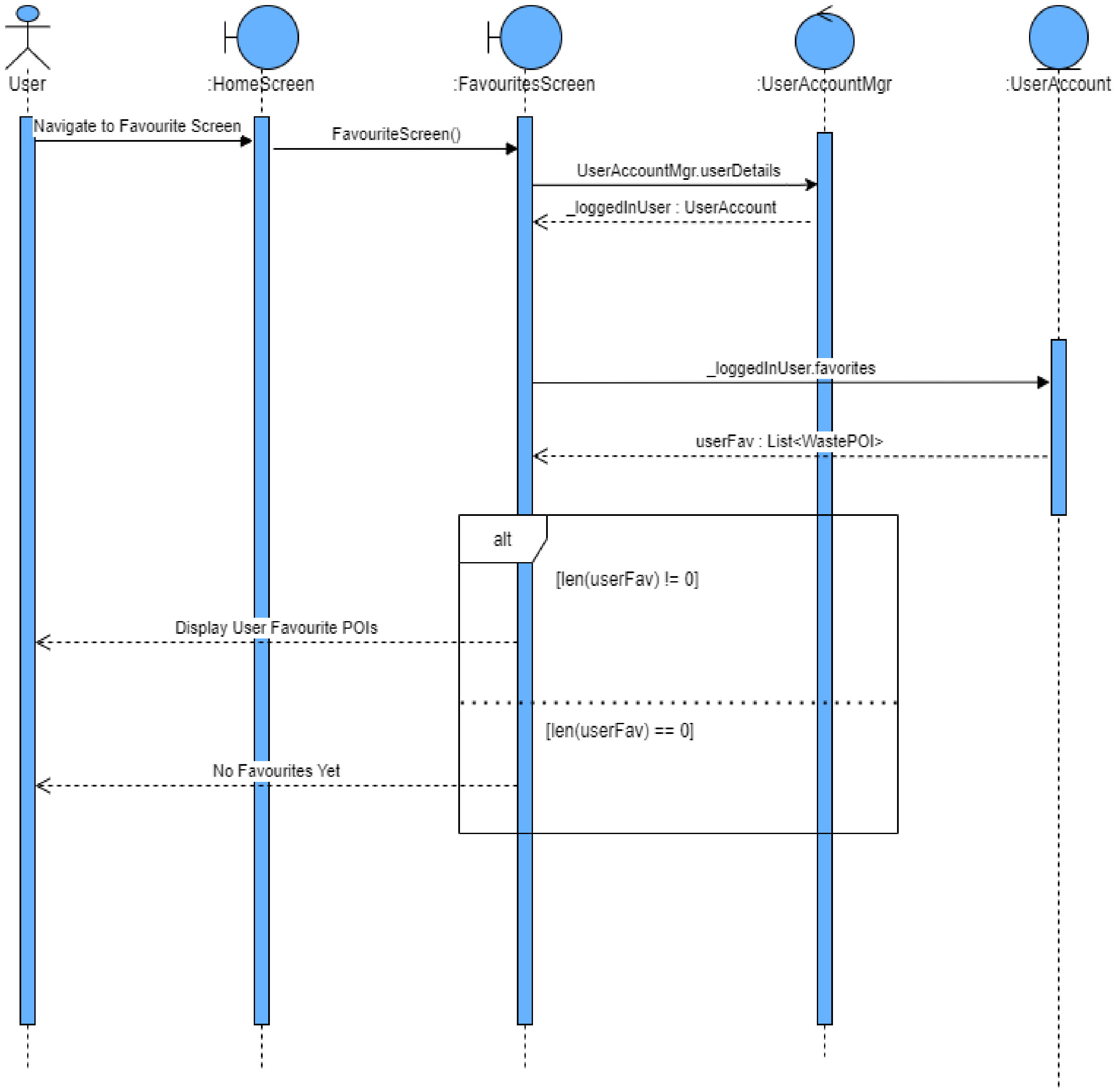
4.8.5. Register User


<<user>>
Actor
: SignUpScreen
OTP Screen
: RegistrationMgr
: UserAccountMgr
: OTP Mgr
Name, email, username, password, confirm password
alt
[password != confirmPassword]
Re-enter details
[password == confirmPassword]
validateUsername_Email(username, email)
username_emailExist: String
alt
[username_emailExist == null]
sendOTP(email)
name, email, username, password, SignUp = true
loop
[correctOTP == false && maxLimitReached == false]
Enter OTP
enteredOTP
verifyOTP(enteredOTP)
correctOTP: boolean
alt
[correctOTP == true]
registerUserAccount (name, email, username, password)
Registration Successful
[correctOTP == false]
maxTries()
maxLimitReached: boolean
maxTries() decrements the number of remaining attempts
[username_emailExist != null]
Username/Email already registered
alt
[maxLimitReached == true]
Registration Failed!

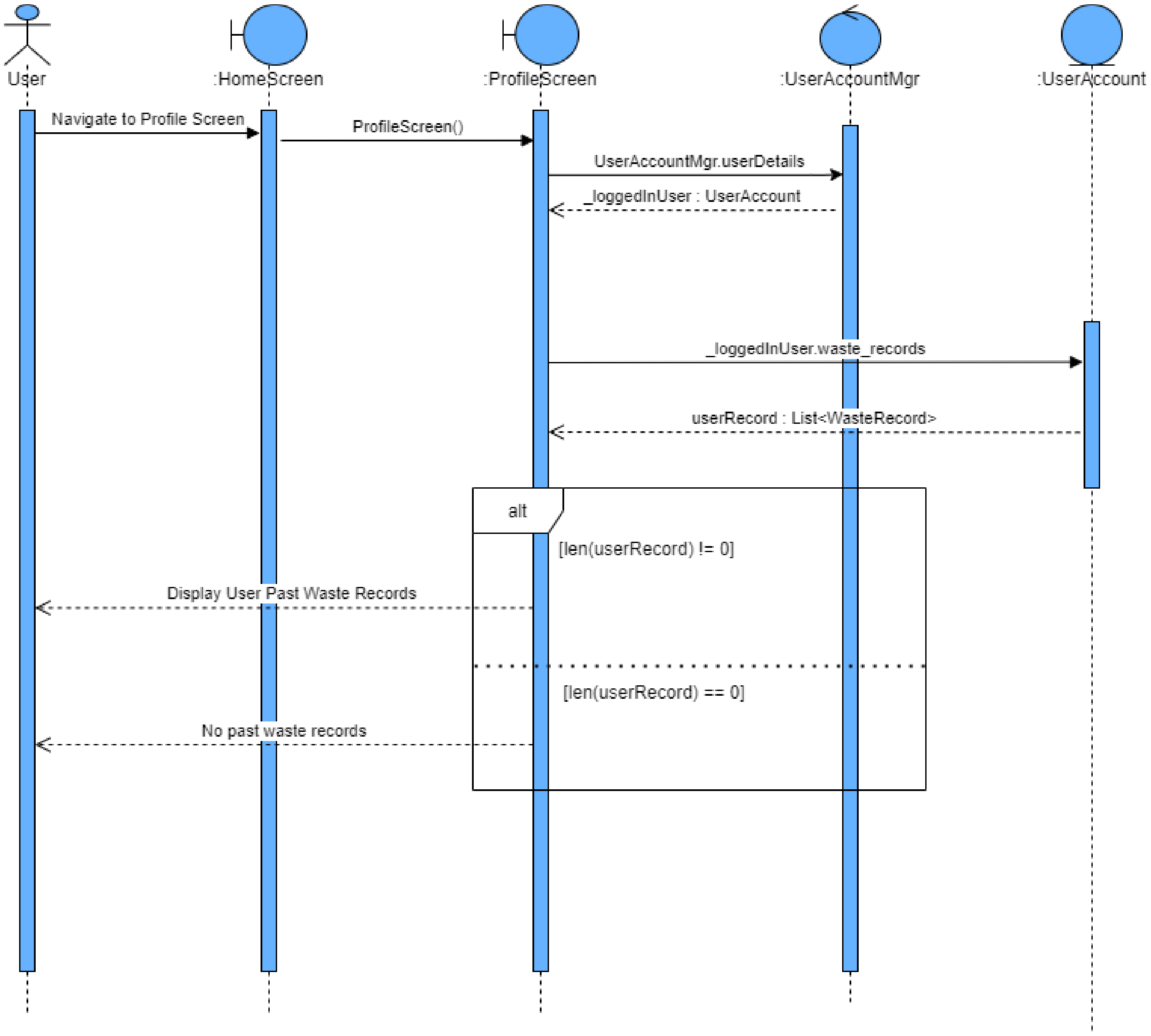
4.8.6. View Catalog



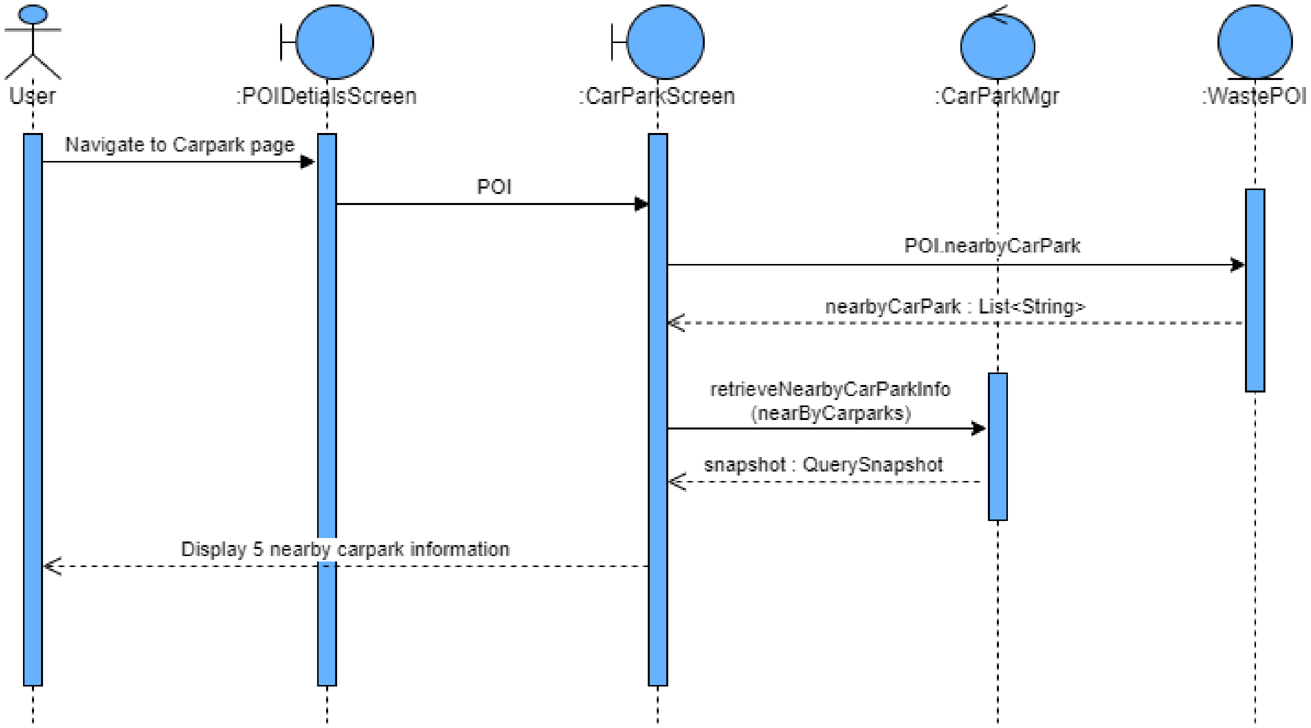
4.8.7. View Favorites



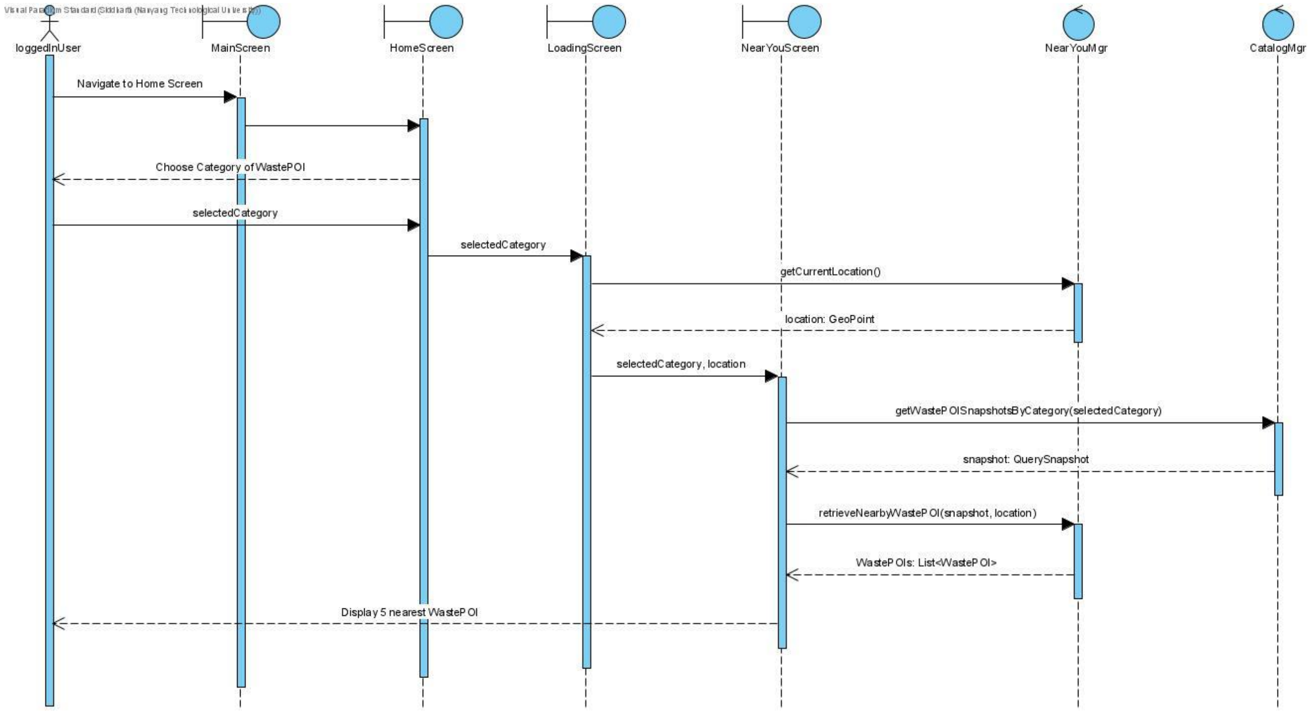
4.8.8. View Past Waste Data Analytics


User
:HomeScreen
:ProfileScreen
:UserAccountMgr
:UserAccount
Navigate to Profile Screen
ProfileScreen()
UserAccountMgr.userDetails
_loggedInUser : UserAccount
_loggedInUser.waste_records
userRecord : List<WasteRecord>
alt
[len(userRecord) != 0]
Display User Past Waste Records
[len(userRecord) == 0]
No past waste records

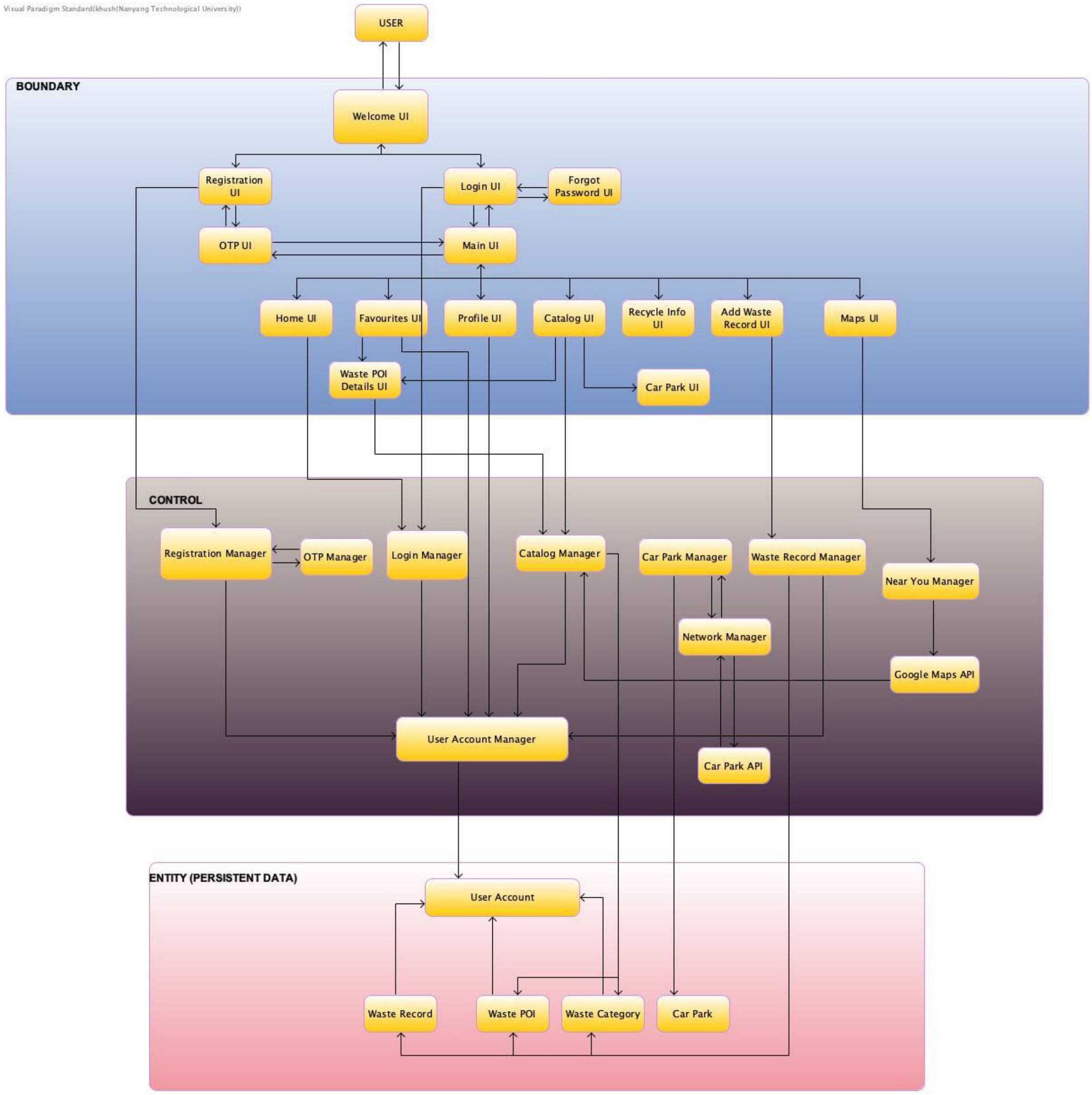
4.8.9. View Car Park Availability near WastePOI



4.8.10. View Nearby WastePOI


Visual Paradigm Standard (Siddharth (Nanyang Technological University))
loggedInUser
MainScreen
HomeScreen
LoadingScreen
NearYouScreen
NearYouMgr
CatalogMgr
Navigate to Home Screen
Choose Category of WastePOI
selectedCategory
selectedCategory
getCurrentLocation()
location: GeoPoint
selectedCategory, location
getWasteP OISnapshotsByCategory(selectedCategory)
snapshot: QuerySnapshot
retrieveNearbyWasteP OI(snapshot, location)
WasteP OIs: List<WasteP OI>
Display 5 nearest WasteP OI

4.9.ARCHITECTURE DIAGRAM



Rationale behind choosing this architecture style

Wastetastic has adapted to the layered subtype of the call-and-return architectural style. This is because the aforementioned architecture segregates the program into departments where each layer looks after its set of functions, processes and controls.

The Control layer of the architectural model represents all the classes which manage the main functionalities of our application including API calls and data retrieval and updating.

The third layer of the architecture- Entity (Persistent Data) sits below the Control layer. It comprises of the User Database, Waste Record Database, Waste POI Database, Car Park Database and Waste Category Database. The User Database stores all the personal details of the user and is directly related to the Waste Record database which stores the waste disposal history of each and every user. The Waste POI database contains the details of each waste disposal and recycling point available in Singapore. It contains the waste POIs for each waste category mentioned in the Waste Category database. The Car Park database stores the details of parking space present in Singapore so that the user can find nearby car parks from the waste POIs.

The Boundary layer interacts directly with the user and provides the input to the Control layer. It provides a level of abstraction since it faces the user while the Control and Persistent Data layers act as the backend of the application.

4.10. DESIGN PATTERNS

The strategy pattern, factory pattern and observer patterns were implemented in the making of this application. The application can be divided into three different levels- Entity, Control and Boundary (Screens). The factory pattern is seen in the use of the different entity classes. For example, the UserAccount Entity class allows for the creation of separate objects for each user which interacts with the control classes separately. These control classes are accessed by the boundary classes and as a result, loose coupling is enabled. The boundary classes contain the entire UI of the application and allows for easy changes, even during runtime. This also prevents the UI of the application to have any interaction with the functionality of the application. This loose coupling between the UI and the functionality and data of the application shows the Observer Pattern implemented by our application. The control classes implement the functionality of the application and interact with entity classes as well. Thus, the strategy pattern is exhibited using this structure.

OTHER NONFUNCTIONAL REQUIREMENTS

5.1.Usability Requirements

5.1.1. The system must display its contents in English.

5.1.1.1. To allow the user to avoid the language barrier as English is the commonly used language in the world.

5.1.1.2. The system must be easily navigable.

5.1.1.2.1. To keep the display simple and allow the user to browse through the app without any issues.

5.1.1.3. The system must offer information about waste management and help assist them in navigating through it.

5.1.1.3.1. To display information about waste management, how to recycle different types of waste and how to dispose of waste to the waste collector.

5.1.1.3.2. To provide help which will contain necessary details on different features of the app and how to navigate through each of them.

5.1.1.4. The system must maintain consistency.

5.1.1.4.1. A consistent visual layout must be adopted throughout the app (e.g., headers, app content, symbols, fonts, colour)

5.1.2. Reliability Requirements

5.1.2.1. The system must retrieve the most approximate GPS location of the user's device, nearest waste collector and the car parking nearest to the selected waste collector.

5.1.2.2. The system must have the latest information to keep the user updated whenever the app is used.

5.1.2.3. The system will implement the Secure Hash Algorithm to perform salt hashing on the passwords before storing it into a database to mask the password field in order to prevent any potential threat to the security of the user.

5.1.3. Performance Requirements

5.1.3.1. The system must not crash whenever the user uses the app.

5.1.3.2. The system must consistently give the desired result in a flick of a second.

5.1.3.3. The system must display appropriate errors if occurred at any point of time.

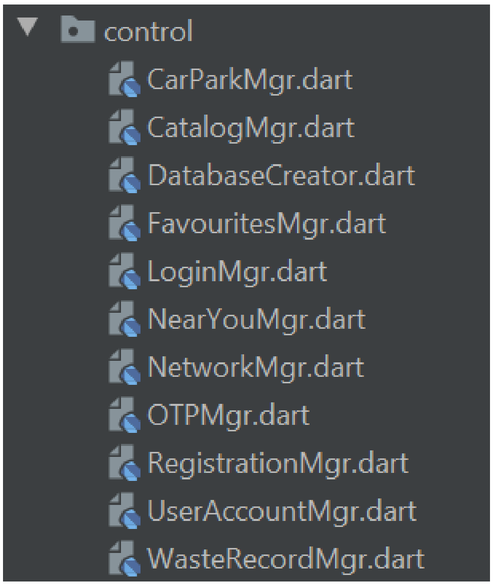
5.1.3.4. The system must allow users to utilize the features (e.g., adding waste record, logging in, switching between screens, etc.) seamlessly and without any delay.

5.1.4. Supportability Requirements

5.1.4.1. Users on both Android and iOS devices must be able to use the app.

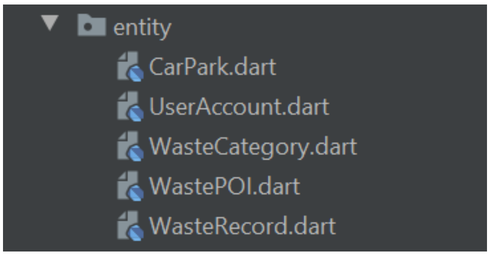
APPLICATION SKELETON

6.1.Control



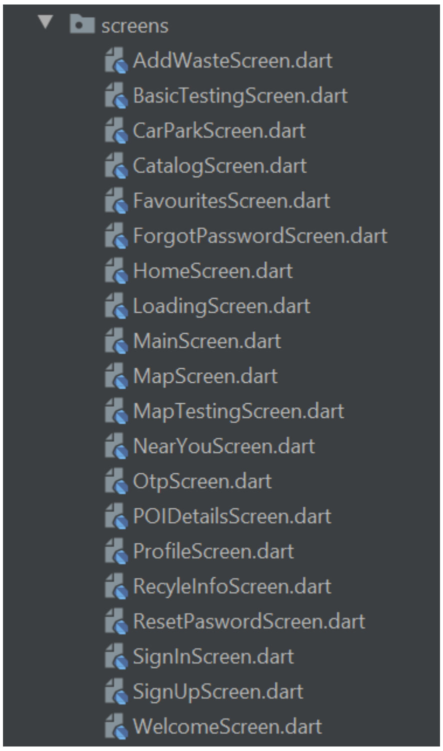
```
▼ control
    CarParkMgr.dart
    CatalogMgr.dart
    DatabaseCreator.dart
    FavouritesMgr.dart
    LoginMgr.dart
    NearYouMgr.dart
    NetworkMgr.dart
    OTPMgr.dart
    RegistrationMgr.dart
    UserAccountMgr.dart
    WasteRecordMgr.dart
```

6.2.**Entity**



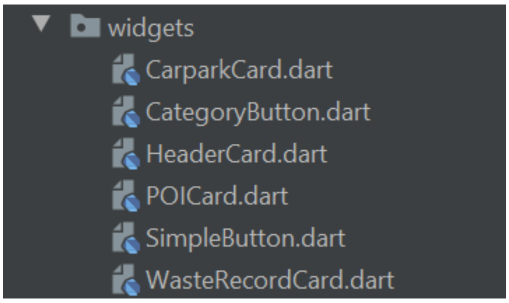
```
▼ entity
    CarPark.dart
    UserAccount.dart
    WasteCategory.dart
    WastePOI.dart
    WasteRecord.dart
```

6.3.Boundary



```
▼ screens
    AddWasteScreen.dart
    BasicTestingScreen.dart
    CarParkScreen.dart
    CatalogScreen.dart
    FavouritesScreen.dart
    ForgotPasswordScreen.dart
    HomeScreen.dart
    LoadingScreen.dart
    MainScreen.dart
    MapScreen.dart
    MapTestingScreen.dart
    NearYouScreen.dart
    OtpScreen.dart
    POIDetailsScreen.dart
    ProfileScreen.dart
    RecyleInfoScreen.dart
    ResetPaswordScreen.dart
    SignInScreen.dart
    SignUpScreen.dart
    WelcomeScreen.dart
```

6.4.Widgets



```
▼ widgets
    CarparkCard.dart
    CategoryButton.dart
    HeaderCard.dart
    POICard.dart
    SimpleButton.dart
    WasteRecordCard.dart
```

TEST CASES

7.1.BLACK BOX TESTING:

7.1.1. Login

7.1.1.1. Generic Cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Correct Login Credentials	Login with valid username and password	The system redirects the user to the home screen	The system redirects the user to the home screen
2.	Incorrect Login Credentials	Login with invalid username and password	The system displays an error message and prompts the user to re-enter details	The system displays an error message and prompts the user to re-enter details
3.	Empty Fields	Login without filling up username or password	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields

7.1.1.2. Specific Cases

Username	Password	Expected Result	Actual Result
User123	123456	Successful Login	Successful Login
Empty(“”)	123456	Please fill in all fields	Please fill in all fields
User123	Empty(“”)	Please fill in all fields	Please fill in all fields
WrongUser	123456	Entered username does not exist in database	Entered username does not exist in database

7.1.2. Sign Up

7.1.2.1. Generic Cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Correct Registration	Sign up with valid username and	The system redirects the user to the home screen	The system redirects the user to the home screen

		password, and user enters correct OTP		
2.	Empty Fields	Sign up with incomplete fields	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
3.	Password Mismatch	Sign up with mismatched password & confirm password fields	The system displays an error message and prompts the user to matching passwords.	The system displays an error message and prompts the user to matching passwords.
4.	Short Password Length	Sign up with password less than 6 characters long	The system displays an error message and prompts the user to enter a valid password	The system displays an error message and prompts the user to enter a valid password
5.	Username/Email Already Registered	Sign up with existing username/email	The system displays an error message and prompts the user to choose a different username/email	The system displays an error message and prompts the user to choose a different username/email
6.	Email ID does not exist	User enters non-existent email ID	The system displays an error message and prompts the user to enter a correct email ID	The system displays an error message and prompts the user to enter a correct email ID
7.	Incorrect OTP	User enter incorrect OTP (sent to email)	The system displays an error message and prompts the user to try again if 3 tries are not reached, else redirect to sign up page	The system displays an error message and prompts the user to try again if 3 tries are not reached, else redirect to sign up page

7.1.2.2. Specific Cases (Email Address)

Note: Assume that test@gmail.com is a valid existing email ID, and null@gmail.com is a non-existent email ID.

Email ID	Expected Result	Actual Result
test@gmail.com	Approve	Approve
null@gmail.com	Reject	Reject
test	Reject	Reject
test@gmail	Reject	Reject

7.1.2.3. Specific Cases (Registration Details)

Note:

i. In successful case below, it is assumed that the user goes on to enter the correct OTP received at his/her email ID.

ii. Assume user@gmail.com is not a registered email ID.

iii. Assume olduser@gmail.com is a registered email ID.

iv. Assume olduser is a registered username.

Username	Email ID	Name	Password	Confirm Password	Expected Result	Actual Result
user	user@gmail.com	Adam Eve	123456	123456	New user created	New user created
Empty(“”)	user@gmail.com	Adam Eve	123456	123456	Please fill in all fields	Please fill in all fields
user	Empty(“”)	Adam Eve	123456	123456	Please fill in all fields	Please fill in all fields
user	user@gmail.com	Empty(“”)	123456	123456	Please fill in all fields	Please fill in all fields
user	user@gmail.com	Adam Eve	Empty(“”)	123456	Please fill in all fields	Please fill in all fields
user	user@gmail.com	Adam Eve	123456	Empty(“”)	Please fill in all fields	Please fill in all fields
user	user@gmail.com	Adam Eve	123	123	Password must be at 6 characters	Password must be at 6 characters
olduser	user@gmail.com	Adam Eve	123456	123456	Username already registered	Username already registered
user	olduser@gmail.com	Adam Eve	123456	123456	Email ID already registered	Email ID already registered
user	user@gmail.com	Adam Eve	123456	123457	Passwords do not match	Passwords do not match

7.1.2.4. Specific Cases (OTP)

Note: Assume that the user has entered all correct details at the sign up page and receives the OTP. Assume the correct OTP is 246810.

Entered OTP	No. of remaining tries	Expected Result	Actual Result
246810	3	The system confirm registration and redirects user to home screen	The system confirm registration and redirects user to home screen
123456	3	The system displays an error message and prompts the user to enter the correct OTP in 2 more tries	The system displays an error message and prompts the user to enter the correct OTP in 2 more tries
123456	1	The system displays an error message and redirects the user to the sign up page	The system displays an error message and redirects the user to the sign up page

7.1.3. Forgot Password

7.1.3.1. Generic Cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Successful Reset	Reset password with correct registered email, and correct OTP	The system changes user’s password and redirects the user to the login screen	The system changes user’s password and redirects the user to the login screen
2.	Empty fields	User leaves fields incomplete	The system prompts the user to fill up the required fields	The system prompts the user to fill up the required fields
3.	Email not registered	User enters email which is not registered	The system displays an error message and prompts the user to re-enter the email ID.	The system displays an error message and prompts the user to re-enter the email ID.

7.1.3.2. Specific Cases (Email Address)

Email ID	Expected Result	Actual Result
test@gmail.com	Approve	Approve
null@gmail.com	Reject	Reject
test	Reject	Reject
test@gmail	Reject	Reject

7.1.4. Add Waste Record

7.1.4.1. Generic Cases

Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Correct Record Details	Add record with correct date, time, weight, and waste category	The system stores waste record and shows a popup - New Record Added	The system stores waste record and shows a popup - New Record Added
2.	Incorrect Fields	Entering invalid fields- date, time, weight	The system displays an error message and prompts the user to re-enter details	The system displays an error message and prompts the user to re-enter details
3.	Empty Fields	Entering record with incomplete fields	The system prompts the user to fill up the required fields	The system uses default values to fill up empty fields

7.1.4.2. Specific Cases (Date and Time)

Note:

i. Assume that the current date is 31st March, 2021, and time is 12:00PM.

ii. The default date taken is the current date and the time taken is 12:00 PM.

Date	Time	Expected Result	Actual Result
Mar 30, 2021	12:00	Accept	Accept
Empty(“”)	12:00	Please fill in the empty fields	Accept- system takes in the default date
Mar 30, 2021	Empty(“”)	Please fill in the empty fields	Accept- system takes in the default time
Apr 10, 2021	12:00	Entered date and time pair is not valid, prompt user to re-enter	Entered date and time pair is not valid, prompt user to re-enter
Mar 31, 2021	13:00	Entered date and time pair is not valid, prompt user to re-enter	Entered date and time pair is not valid, prompt user to re-enter
Mar 41, 2021	12:00	Entered date is not valid, prompt user to re-enter	Entered date is not valid, prompt user to re-enter
Mar 30, 2021	32:00	Entered time is not valid, prompt user to re-enter	Entered time is not valid, prompt user to re-enter

7.1.4.3. Specific Cases (Weight)

Note: The unit of weight is taken as kg by default.

Note: The default weight taken is 7kg.

Weight	Expected Result	Actual Result
7	Accept	Accept
Empty(“”)	Please fill in the empty field	Accept- system takes in the default date
0	Entered weight is not valid, prompt user to re-enter	Entered weight is not valid, prompt user to re-enter (to change code)

7.1.5. Favourites

Test ID	Test Name	Scenario	Expected Result	Actual Result
1.	Add Favourite	Favourite a Waste POI	The system saves the POI into the User’s favourite list	The system saves the POI into the User’s favourite list
2.	Remove Favourite	Favourite an existing Waste POI	The system removes the POI from the User’s favourite list	The system removes the POI from the User’s favourite list

7.1.6. Display POIs near the User

7.1.6.1. Generic Cases

Test ID	Location	Expected Result	Actual Result
1.	Location enabled and allowed for app	Display 5 Waste POIs near current location	Display 5 Waste POIs near current location

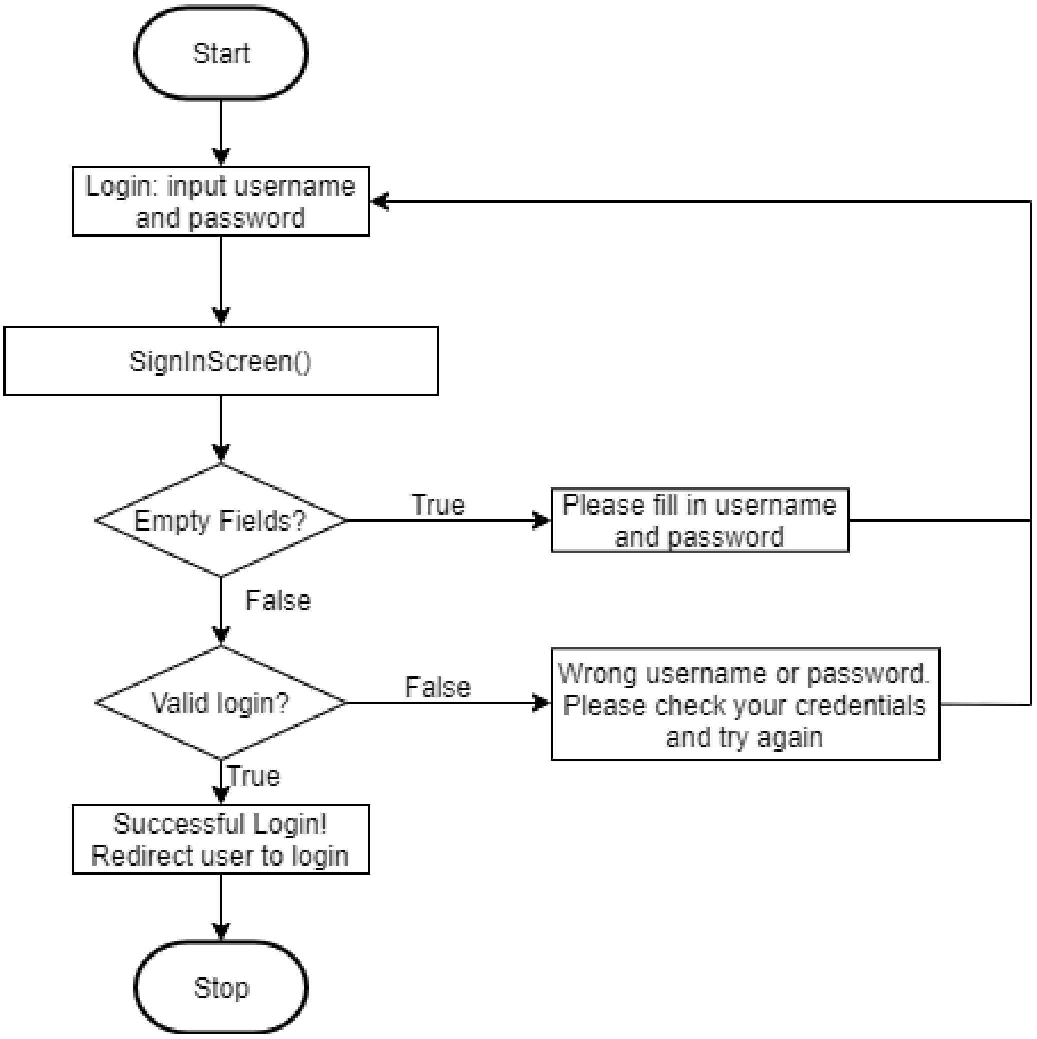
2.	Location disabled	Display error message “Location Unavailable”	Display error message “Location Unavailable”
3.	Location enabled but not allowed for app	Prompt user for location permission	Prompt user for location permission

7.1.6.2. Specific Cases (Different Locations)

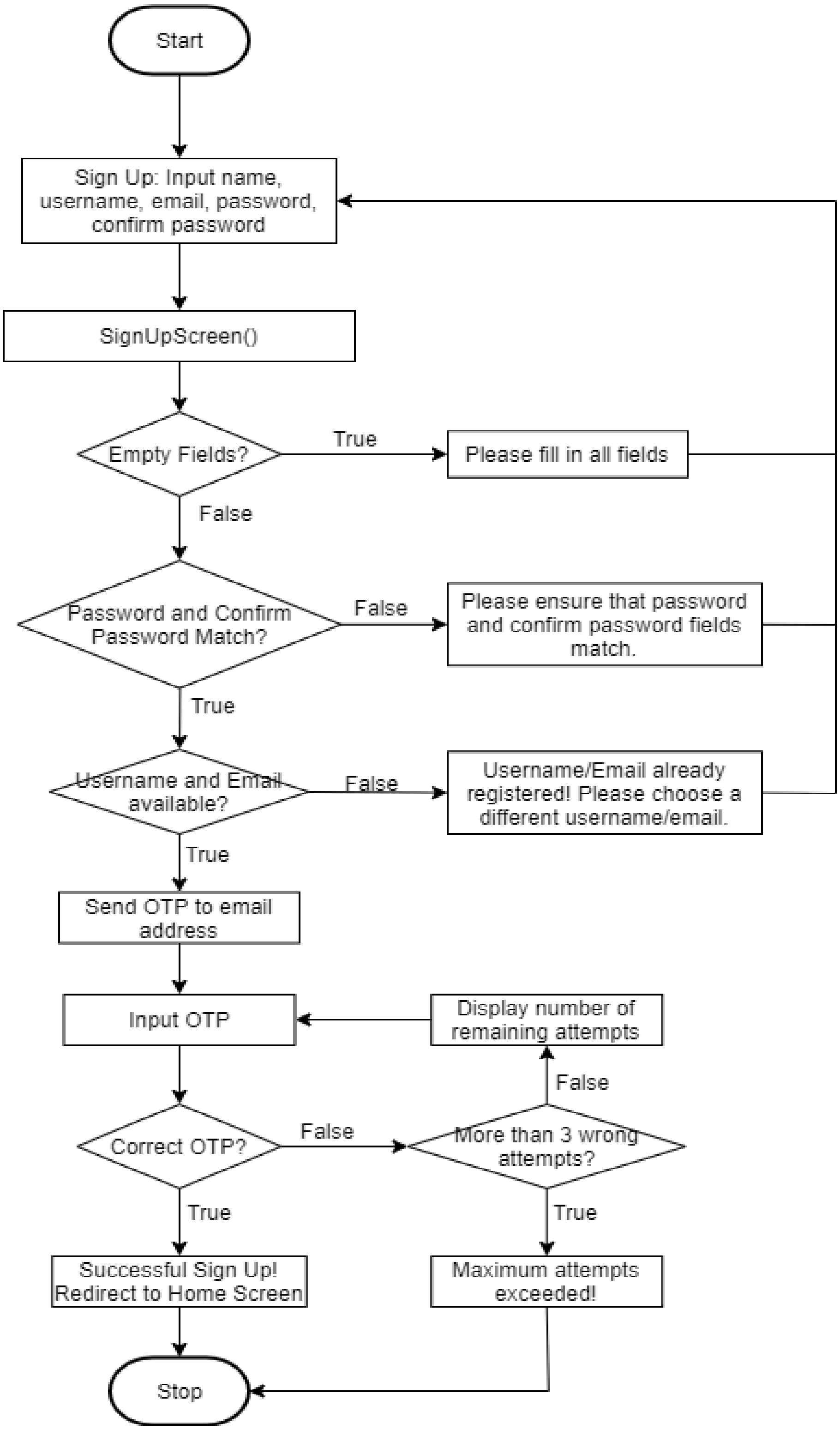
Latitude	Longitude	Expected Result	Actual Result
1.434145	103.786604	Display 5 Waste POIs near Woodlands Ave 5	Display 5 Waste POIs near Woodlands Ave 5
1.34472	103.68139	Display 5 Waste POIs near NTU	Display 5 Waste POIs near NTU
1.3343028	103.85633	Display 5 Waste POIs near Toa Payoh	Display 5 Waste POIs near Toa Payoh

7.2. WHITE BOX TESTING

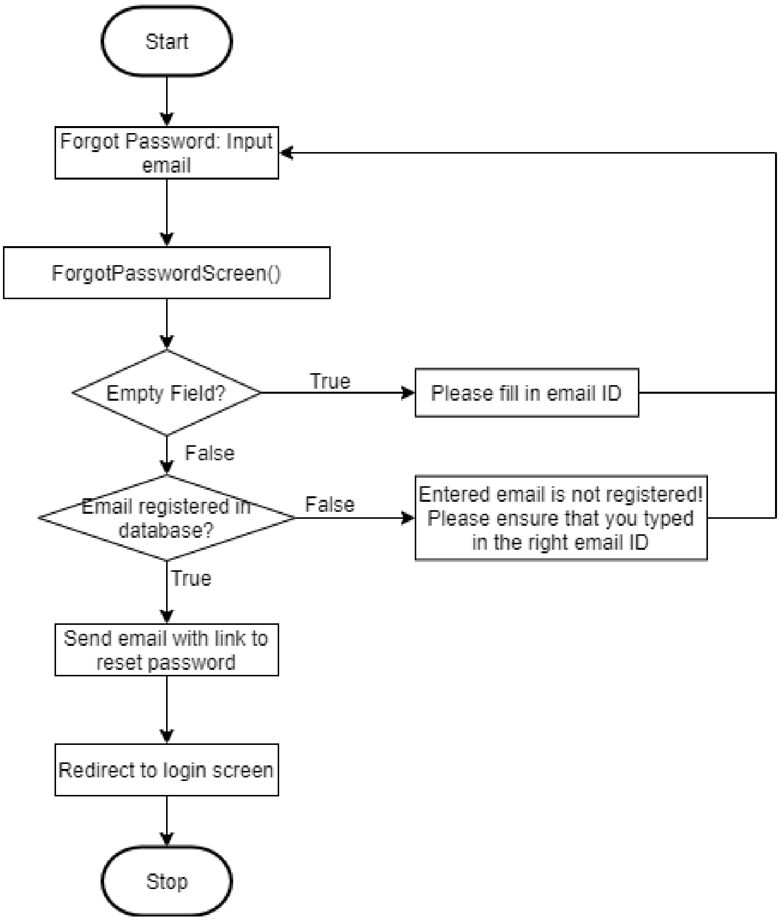
7.2.1. Login


Start
Login: input username and password
SignInScreen()
Empty Fields?
True
Please fill in username and password
False
Valid login?
False
Wrong username or password. Please check your credentials and try again
True
Successful Login! Redirect user to login
Stop

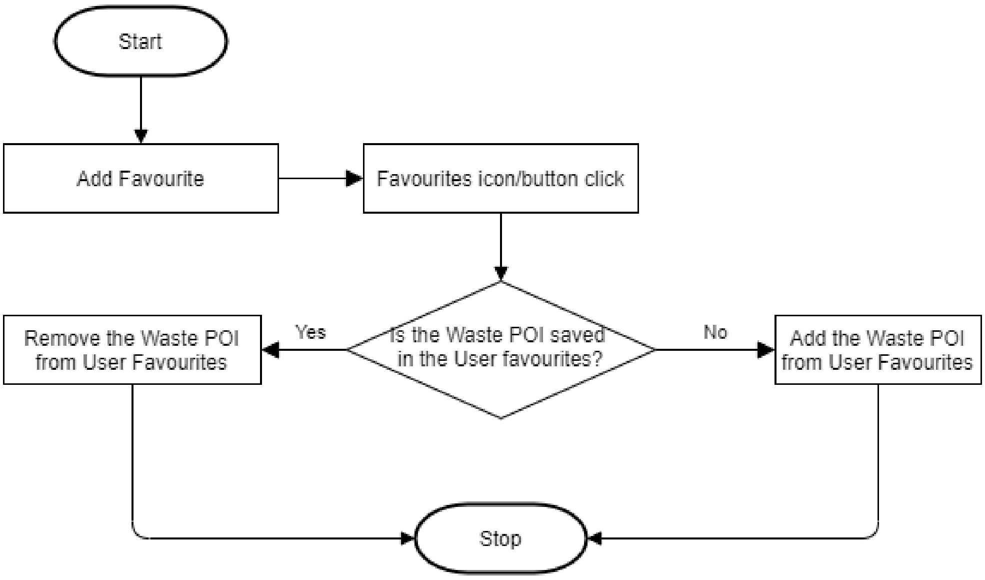
7.2.2. Sign Up


Start
Sign Up: Input name, username, email, password, confirm password
SignUpScreen()
Empty Fields?
True
Please fill in all fields
False
Password and Confirm Password Match?
False
Please ensure that password and confirm password fields match.
True
Username and Email available?
False
Username/Email already registered! Please choose a different username/email.
True
Send OTP to email address
Input OTP
Display number of remaining attempts
False
Correct OTP?
False
More than 3 wrong attempts?
True
True
Successful Sign Up! Redirect to Home Screen
Maximum attempts exceeded!
Stop

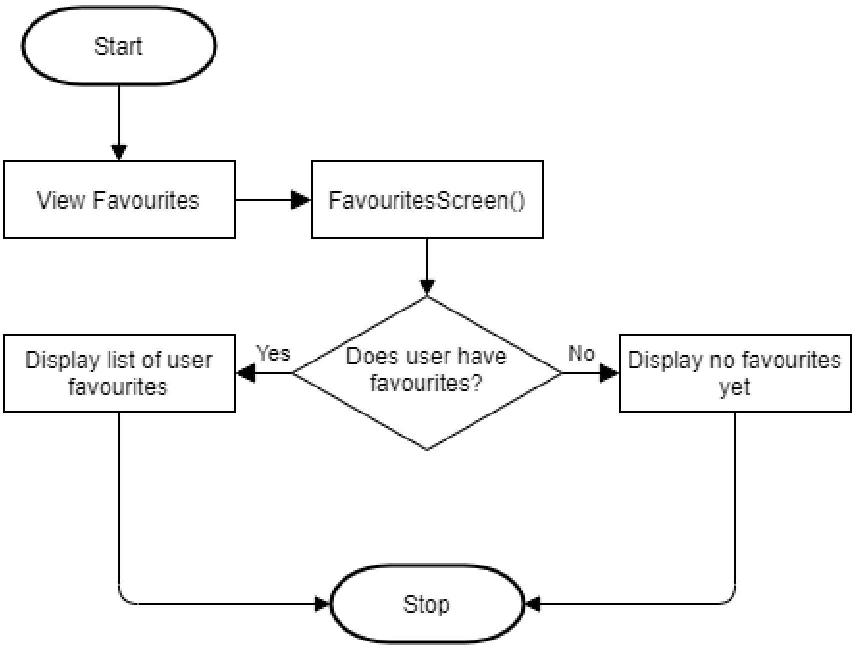
7.2.3. Forgot Password


Start
Forgot Password: Input email
ForgotPasswordScreen()
Empty Field?
True
Please fill in email ID
False
Email registered in database?
False
Entered email is not registered! Please ensure that you typed in the right email ID
True
Send email with link to reset password
Redirect to login screen
Stop

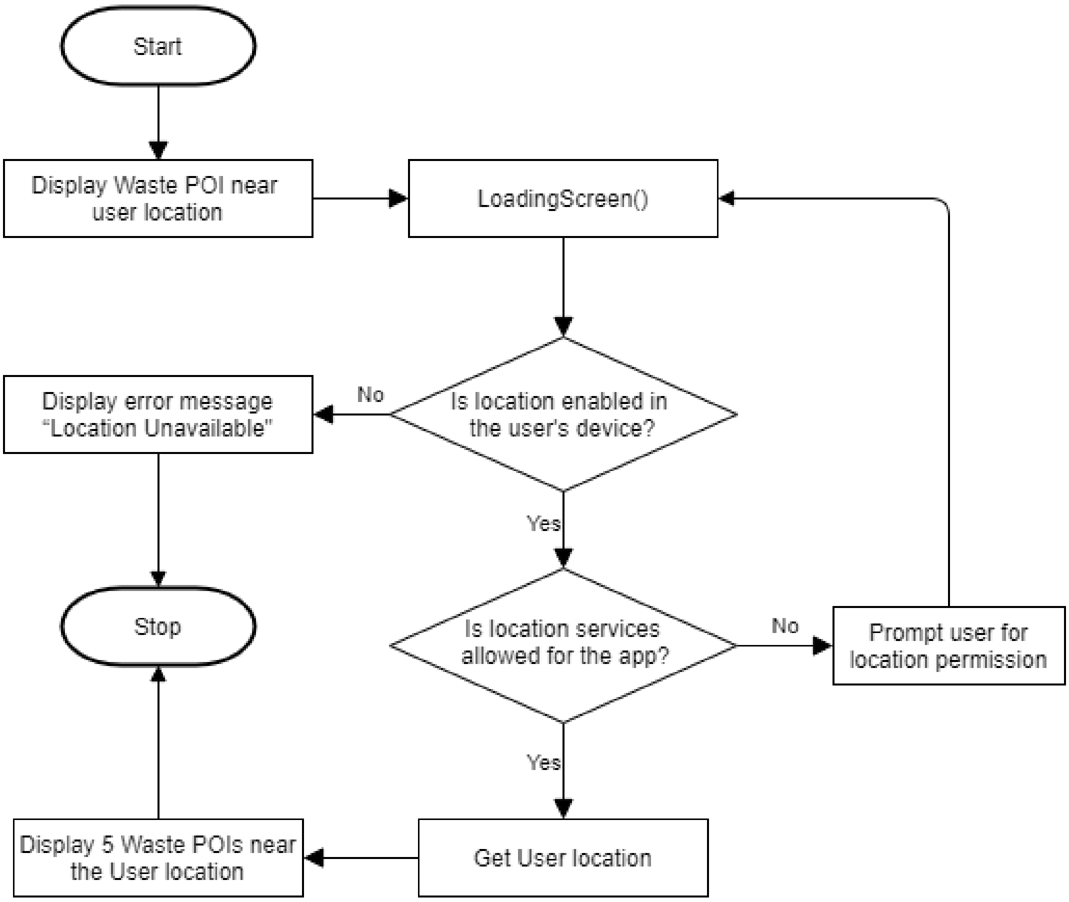
7.2.4. Add Favorites


Start
Add Favourite
Favourites icon/button click
Is the Waste POI saved in the User favourites?
Yes
Remove the Waste POI from User Favourites
No
Add the Waste POI from User Favourites
Stop

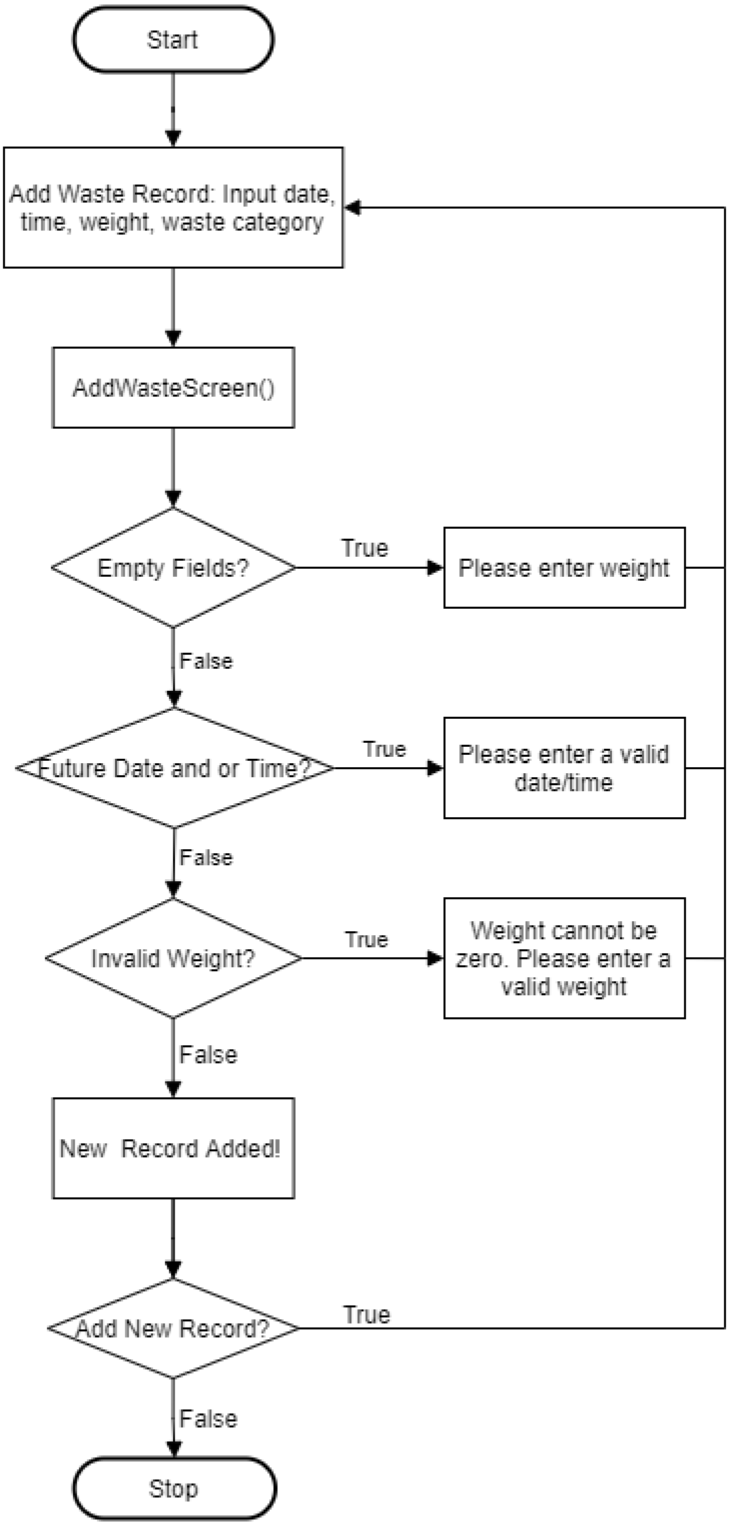
7.2.5. View Favorites


Start
View Favourites
FavouritesScreen()
Does user have favourites?
Yes
No
Display list of user favourites
Display no favourites yet
Stop

7.2.6. View nearby Waste POI


Start
Display Waste POI near user location
LoadingScreen()
Is location enabled in the user's device?
No
Display error message "Location Unavailable"
Yes
Stop
Is location services allowed for the app?
No
Prompt user for location permission
Yes
Get User location
Display 5 Waste POIs near the User location

7.2.7. Add Waste Record



DATA DICTIONARY

Term	Definition
System	Refers to the Waste Management app itself.
User	Refers to the person using the Waste Management app.
Validate	To check the correctness of the input.
Home Screen	The main screen of the app, that can be customized according to the user.
Waste POI	Refers to waste disposal/recycling points within Singapore.
Drop Down Menu	A list which allows the user to choose one value. The list drops down when clicked on to display more options.
Registration	The process of signing up a new user account to use the app.
Database	Collection of all the registered User accounts and corresponding details.
Recycle	The process of converting waste into useful new material or objects.
Google API	Application programming interfaces developed by Google, allows communication with Google Services.
Favorites	An app feature that lets users bookmark their favorite waste collection/recycling points.
Search	An app feature that lets the users search nearby waste collection/recycling points using keywords.
OTP	A One Time Password sent to the user via email for resetting of user password.