
Software Requirements Specification

for

SG!Recycle

Prepared by:

Chew Jie Ying Perlyn, Gao Xinrui,

Justin Yip Jia En, Tan Song Yun

REngineers

6 November 2020

Table of Contents

1. Introduction	4
1.1 Purpose	4
1.2 Document Conventions	4
1.3 Product Scope	4
2. Overall Description	4
2.1 Product Perspective	4
2.2 Product Functions	4
2.3 Operating Environment	4
2.4 User Documentation	4
2.5 Assumptions and Dependencies	5
3. External Interface Requirements	5
3.1 User Interfaces	5
4. Functional Requirements	7
5. Nonfunctional Requirements	7
6. Use Case	7
6.1 Use Case Diagram	7
6.2 Use Case Descriptions	8
6.2.1 Auto-Detection of Current User Location	8
6.2.2 Manual Receipt of Current User Location	8
6.2.3 Returning of Nearest Recycling Bins	9
6.2.4 Dropping Pins on Nearest Recycling Bins	9
6.2.5 Providing Warnings on the Type of Items Recyclable	10
6.2.6 Scanning an Item to Determine if it is Recyclable	10
7. Class Diagrams	11
7.1 Locating Nearest Recycling Bins	11
7.1.1 Stereotyping Class Diagram	11
7.1.2 Interim Class Diagram	11
7.1.3 Final Class Diagram	12
7.2 Recyclable Classifier	12
7.2.1 Stereotyping Class Diagram	12
7.2.2 Interim Class Diagram	12
7.2.3 Final Class Diagram	12
7.3 Class Diagram	13
8. Key Design Issues	13
8.1 Persistent Data/Objects	13
8.2 Access Control	13
9. Sequence Diagrams	14
9.1 LocationFinder	14

9.2	PostalVerifier	14
9.3	ItemScanner	15
10.	<i>Dialog Maps</i>	15
10.1	SG!Recycle	15
10.2	Auto-Locating Nearest Recycling Bins.....	16
10.3	Manual-Locating Nearest Recycling Bins	16
10.4	Classifying User's Item as Recyclable or Not.....	17
11.	<i>System Architecture</i>	17
12.	<i>Testing</i>	18
12.1	Equivalence Class Testing for Manual Location Detection.....	18
12.2	Boundary Value Testing for Manual Location Detection	18
12.3	Combined Test Cases for Manual Location Detection.....	18
12.4	Control Flow Testing for Distance Calculator	19
12.5	Control Flow Testing for Item Classifier	19

1. Introduction

1.1 Purpose

This SRS covers the entire product system.

1.2 Document Conventions

Software Requirements that hold higher priority will be given a preceding index under Functional Requirements.

This documentation is intended for all potential clients interested in implementing SG!Recycle for end-users in Singapore.

1.3 Product Scope

This application aims to make recycling a more convenient task for Singaporeans. By educating Singaporeans on recyclables, automatically detecting the nearest recycling bins to the user, then suggesting if the user's items are recyclable, we cover the end-to-end aspects of recycling.

2. Overall Description

2.1 Product Perspective

This product is derived as an extension of the National Environmental Agency's (NEA) Recycling Bins API. Currently, the Recycling Bins are only provided as coordinates and do not show the location of bins relative to the end-user. SG!Recycle exists to close the gap, providing end-users with the closest Recycling Bins to boost convenience and ultimately encouraging a habit of recycling.

2.2 Product Functions

This product has three main functions. Firstly, to suggest the 10 nearest recycling bins to the user. Secondly, to provide information on the types of material recyclable. Lastly, to allow users to scan items to detect if their items are suitable for recycling.

2.3 Operating Environment

SG!Recycle can be installed on mobile phones using Android operating system. Minimum Android version required is Android 5.0 Lollipop. SG!Recycle requires the Android phone to possess a functional camera as well as Wi-Fi.

2.4 User Documentation

Demo videos are provided to assist users in understanding and using the product.

2.5 Assumptions and Dependencies

This SRS assumes that NEA updates its API in Gov.sg regularly, where the location of updated Recycling Bins does not fall out of the application's reach. SG!Recycle also depends on the Flutter TensorFlow (Machine Learning) API, as well as Google Maps' API for auto-location of the user and dropping of pins on the user-interface map.

3. External Interface Requirements

3.1 User Interfaces

The main interface (Figure 1) will hold information on the various types of recyclables, in hopes of educating our target audience on the various recyclable materials and promoting recycling. For the user to easily navigate to different pages on the app, there is a bottom navigation toolbar.

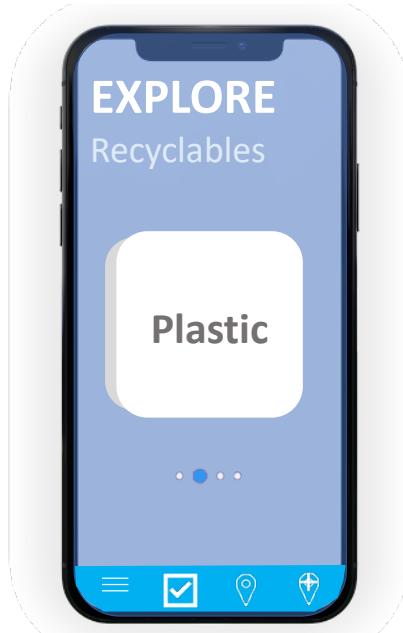


Figure 1: The main interface

The third button brings the user to locate his current location and displays the nearest ten recycling on the map interface using drop pins (Figure 2). The user can pinch to zoom and drag to move the map. In the event that the automatic detection of user location is not allowed, or the user wishes to search for a location beforehand, the fourth button allows the user to input his postal code into the system. When an invalid postal code is entered, an error message will be shown on the screen, prompting the user to re-enter a valid postal code.

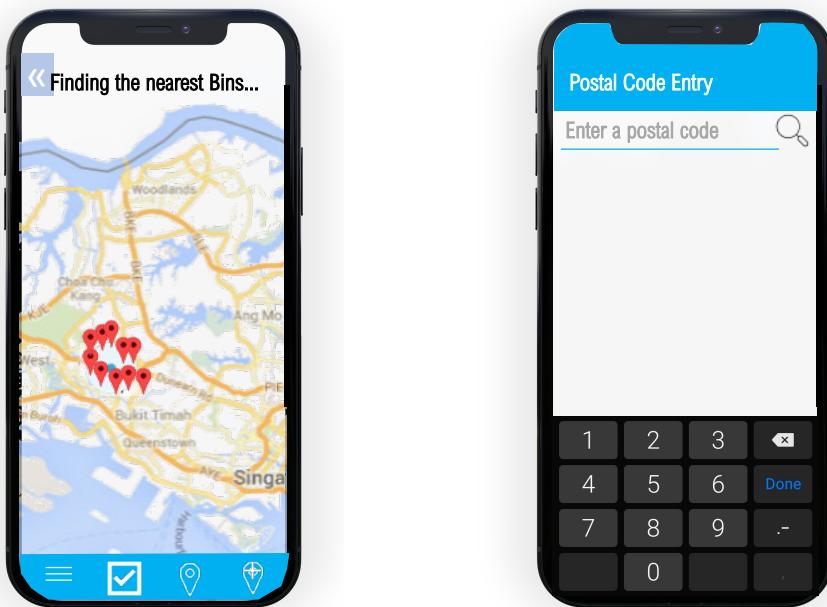


Figure 2: The interfaces to get user's location either automatically or through manual input of postal code

The second button from the left on the toolbar allows the user to check whether his item is recyclable. The user will scan his item through the phone's camera and an indication bar at the bottom will show whether the item is recyclable and the system's confidence (Figure 3).

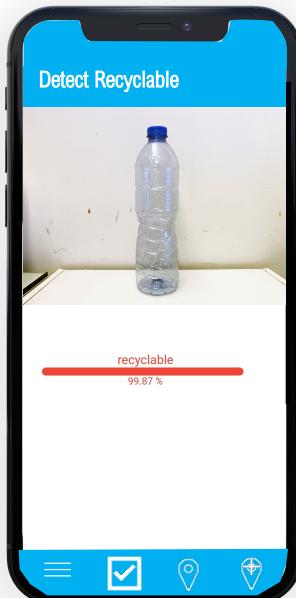


Figure 3: The interface to scan the user's item

4. Functional Requirements

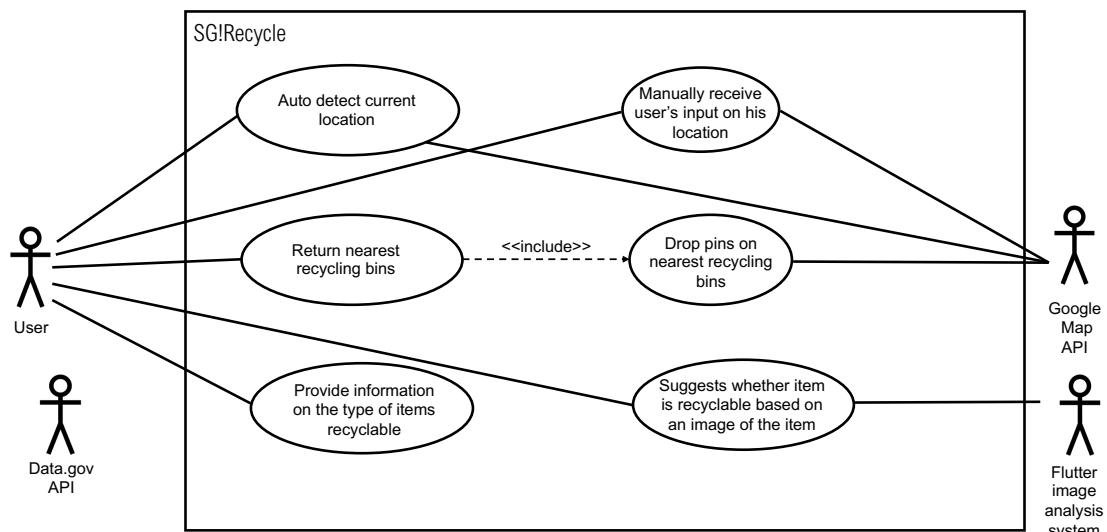
- 1.1 The user must be able to locate his current location on his device when enabled.
- 1.1.1 If the user is unable to locate his current location on his device, he must be able to enter his location.
- 1.2 The system must locate the nearest 10 Recycling Bins to the user, based on his location detected/inputted into the system.
- 1.2.1 The system must convert the user's input address to usable coordinates that can return a pin-pointed value.
- 1.2.2 The system must display the located nearest Recycling Bins via a zoomed-in map of the user's location and drop pins over the location of each Recycling Bin.
- 2.1 The system must provide information on the type of items recyclable.
- 2.1.1 If the user is unsure if his item is recyclable after the warning, the user must be able to scan his object.
- 2.1.2 The system must be able to suggest whether it is recyclable.

5. Nonfunctional Requirements

1. The system must have at least 90% availability.
2. Upon a failure, the application must take less than 1 minute to recover and restart the application.
3. Upon a failure, the application must be able to display the error in the backend.
4. The application must be able to support at least 50 concurrent users.

6. Use Case

6.1 Use Case Diagram



6.2 Use Case Descriptions

6.2.1 Auto-Detection of Current User Location

Use Case ID:	#1		
Use Case Name:	Auto detect current location		
Created By:	Perlyn	Last Updated By:	Perlyn
Date Created:	1 September 2020	Date Last Updated:	22 September 2020
Actor:	User, Google Maps API		
Description:	Use case is to detect the user's current location.		
Preconditions:	User's location services must be enabled.		
Postconditions:	The coordinates of the user's location must be generated.		
Priority:	Highest		
Frequency of Use:	Very frequent		
Flow of Events:	1. The User clicks on the 'Location' button in the bottom navigation bar in the Home Page. 2. The System uses the built-in GPS system to locate the user. 3. The System calls the Google Maps API to pass in the coordinates of the current location of user.		
Extended By:	Manually receive user's input on his location if auto-detection is unsuccessful.		
Includes:	-		
Special Requirements:	User's location services must be enabled		
Assumptions:	The Google Maps API installed is the latest version		
Notes and Issues:	NA		

6.2.2 Manual Receipt of Current User Location

Use Case ID:	#2		
Use Case Name:	Manually receive user's input on his location.		
Created By:	Justin	Last Updated By:	Justin
Date Created:	8 September 2020	Date Last Updated:	22 September 2020
Actor:	User, Google Maps API		
Description:	Use Case is to accept the user's manually inputted location.		
Preconditions:	User must opt to manually input his/her location in the Application, when searching for Recycling Bins.		
Postconditions:	The System uses the manually inputted location to generate its coordinates, then to search for the nearest 10 Recycling Bins.		
Priority:	High		
Frequency of Use:	Frequent		
Flow of Events:	1. The User clicks on the 'Postal Code' button in the bottom navigation bar of the Home Page. 2. A page appears with a text box, allowing him to key in the postal code/location name for detection. 3. The User keys in his location and it is verified against a database of valid postal codes in Singapore. 4. If the postal code is valid, send the location to the Google Maps API.		
Extended by:	-		
Includes:	The Google Maps API drops pins on the Application Map, corresponding to the returned coordinates.		
Special Requirements:	-		
Assumptions:	-		
Notes and Issues:	The user can always opt to retry auto-detection of his/her location instead of manually keying in his location or address.		

6.2.3 Returning of Nearest Recycling Bins

Use Case ID:	#3		
Use Case Name:	Return nearest Recycling Bins.		
Created By:	Justin	Last Updated By:	Justin
Date Created:	8 September 2020	Date Last Updated:	22 September 2020
Actor:	User, Data.gov API		
Description:	Use case is to return the nearest Recycling Bins based on the user's location		
Preconditions:	The user's location must be detected by the System.		
Postconditions:	The coordinates of the Recycling Bins must be passed back as data into the System.		
Priority:	Highest		
Frequency of Use:	Very frequent		
Flow of Events:	1. The System detects the User's locations. 2. The System calls Data.gov to retrieve coordinates of recycling bins. 3. The System uses the User's location coordinates to calculate the distance to the closest bins. 4. The System returns the coordinates of the nearest 10 Recycling Bins.		
Extended by:	-		
Includes:	The Google Maps API drops pins on the Application Map, corresponding to the returned coordinates.		
Assumptions:	-		

6.2.4 Dropping Pins on Nearest Recycling Bins

Use Case ID:	#4		
Use Case Name:	Drop pins on nearest recycling bins.		
Created By:	Song Yun	Last Updated By:	Song Yun
Date Created:	8 Sept 2020	Date Last Updated:	22 Sept 2020
Actor:	Google Map API		
Description:	Use case is to drop pins to indicate the coordinates of the nearest recycling bins.		
Preconditions:	Coordinates of the nearest recycling bins are obtained.		
Postconditions:	Pins appear at the locations on the application map corresponding to the locations of the nearest recycling bins.		
Priority:	Highest		
Frequency of Use:	Very frequent		
Flow of Events:	1. The System receives the coordinates of the nearest recycling bins from the Google Map API. 2. Google Map APIs must drop the pins at the respective coordinates on the application map.		
Extended by:	-		
Includes:	-		
Special Requirements:	-		
Assumptions:	Google API returns the correct coordinates of the nearest recycling bins.		
Notes and Issues:	-		

6.2.5 Providing Warnings on the Type of Items Recyclable

Use Case ID:	#5		
Use Case Name:	Provide a warning on the type of items recyclable		
Created By:	Song Yun	Last Updated By:	Song Yun
Date Created:	8 September 2020	Date Last Updated:	22 September 2020
Actor:	User		
Description:	The System will provide a list of recyclable and non-recyclable items to the user.		
Preconditions:	The user's device must have an active internet connection.		
Postconditions:	The System must output suggestions on the type of items that are recyclable.		
Priority:	Highest		
Frequency of Use:	Very frequent		
Flow of Events:	1. The User clicks on a 'Recyclable' card in the Home Screen. 2. The System displays information on the recyclable type that was chosen		
Extends:	-		
Includes:	-		
Special Requirements:	-		
Assumptions:	-		
Notes and Issues:	-		

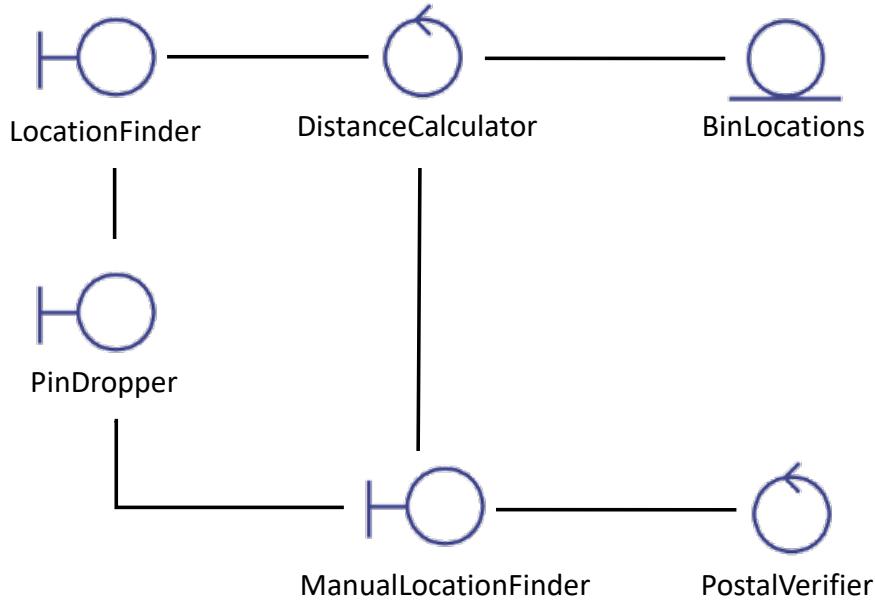
6.2.6 Scanning an Item to Determine if it is Recyclable

Use Case ID:	#6		
Use Case Name:	Scan item to determine if it is recyclable.		
Created By:	Xinrui	Last Updated By:	Xinrui
Date Created:	8 September 2020	Date Last Updated:	8 September 2020
Actor:	User, Machine learning model within mobile app.		
Description:	User will place the potential recyclable within camera view, and the Machine Learning model within the mobile app will determine whether the item is recyclable.		
Preconditions:	The user has allowed the system to use his/her device's camera under his device settings.		
Postconditions:	The system returns the item type and determines if it is recyclable.		
Priority:	High		
Frequency of Use:	Frequent (When the user wishes to clarify whether the item is recyclable).		
Flow of Events:	1. The User clicks on "Verify Recyclables" button on the bottom navigation bar in the Home Page. 2. The System scans the item and calls the Flutter image recognition system. 3. The System gets the item type from the image recognition system and returns it to the user and alerts user if item is recyclable.		
Extended by:	-		
Assumptions:	The item presented is similar to our Machine Learning model's training images.		

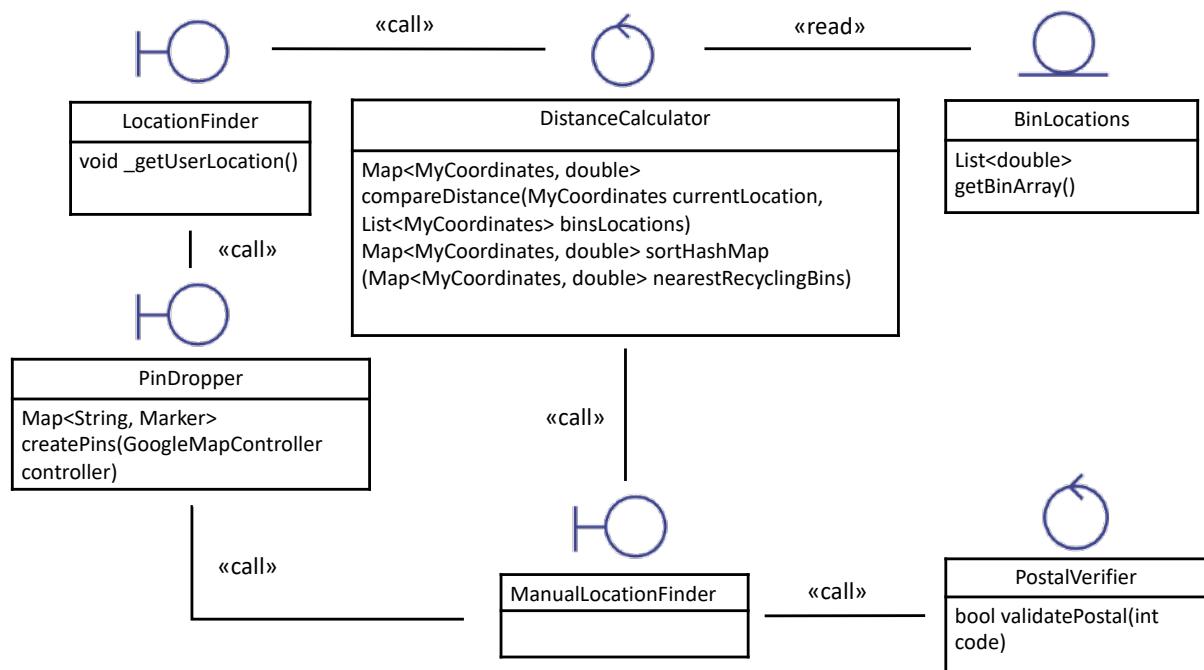
7. Class Diagrams

7.1 Locating Nearest Recycling Bins

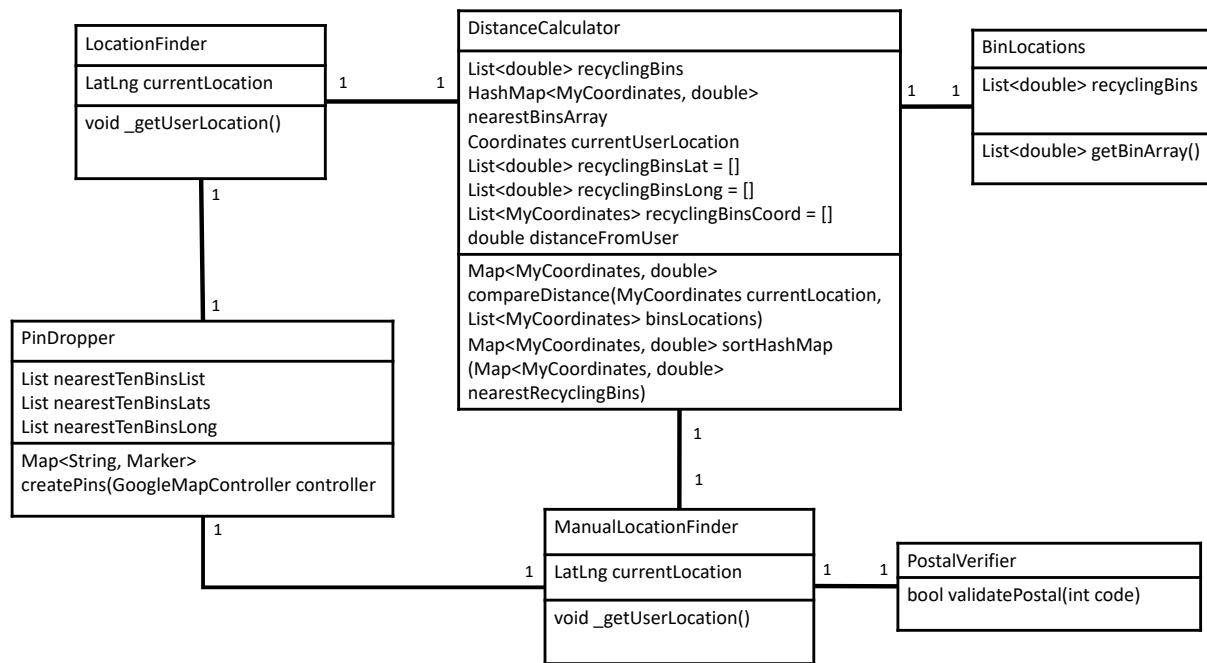
7.1.1 Stereotyping Class Diagram



7.1.2 Interim Class Diagram



7.1.3 Final Class Diagram

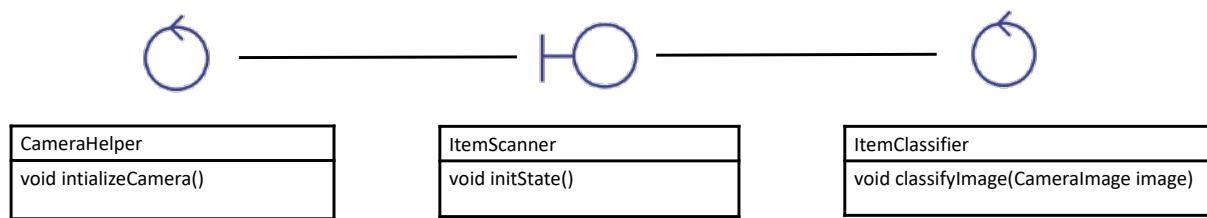


7.2 Recyclable Classifier

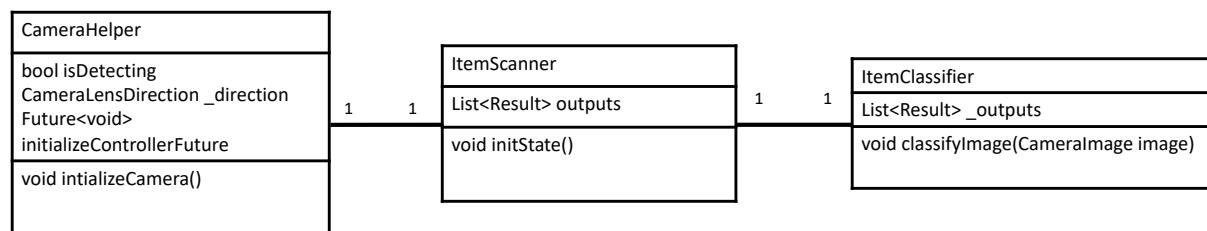
7.2.1 Stereotyping Class Diagram



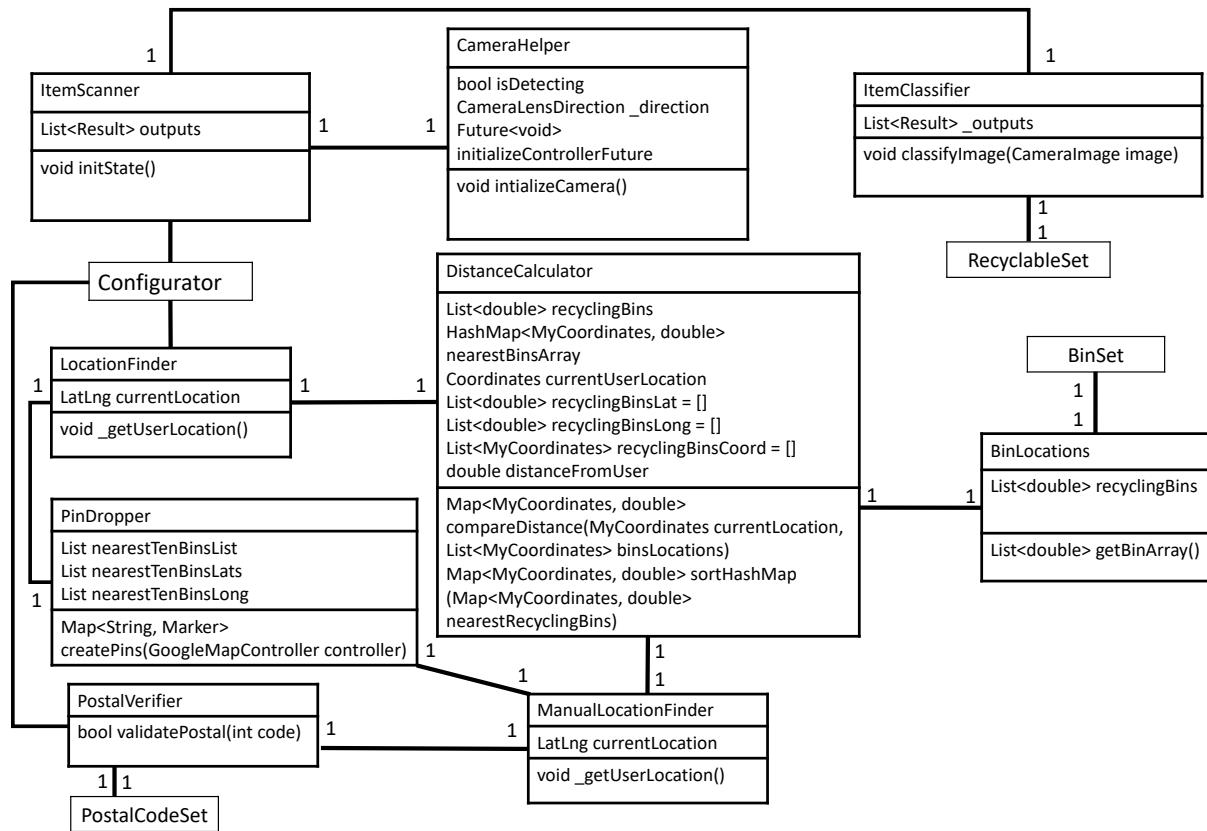
7.2.2 Interim Class Diagram



7.2.3 Final Class Diagram



7.3 Class Diagram



8. Key Design Issues

8.1 Persistent Data/Objects

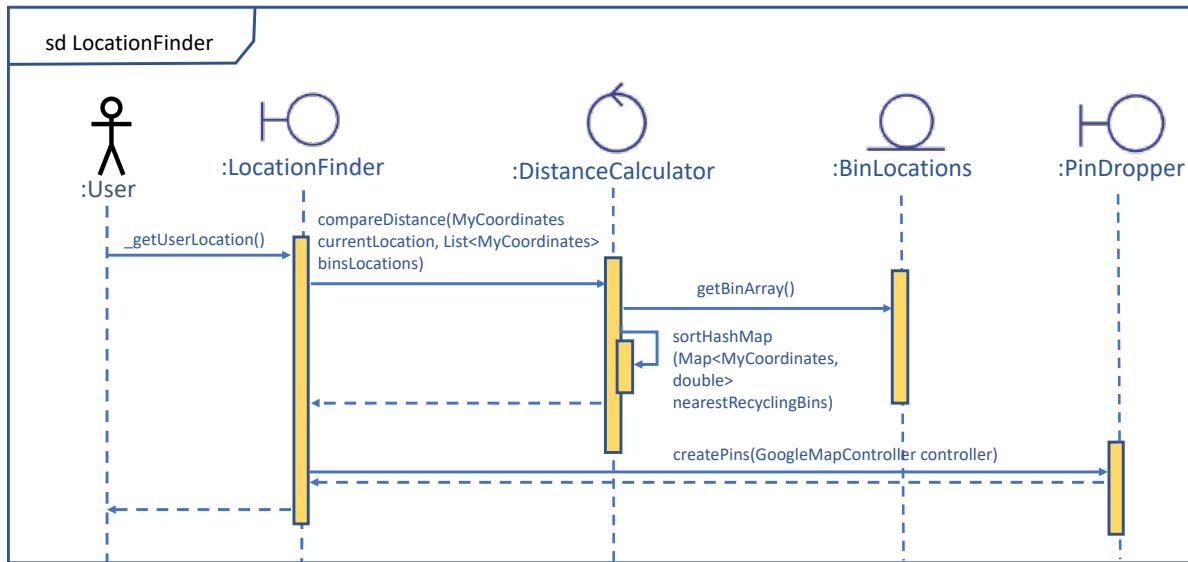
1. BinLocations
2. RecyclableSet
3. PostalCodeSet

8.2 Access Control

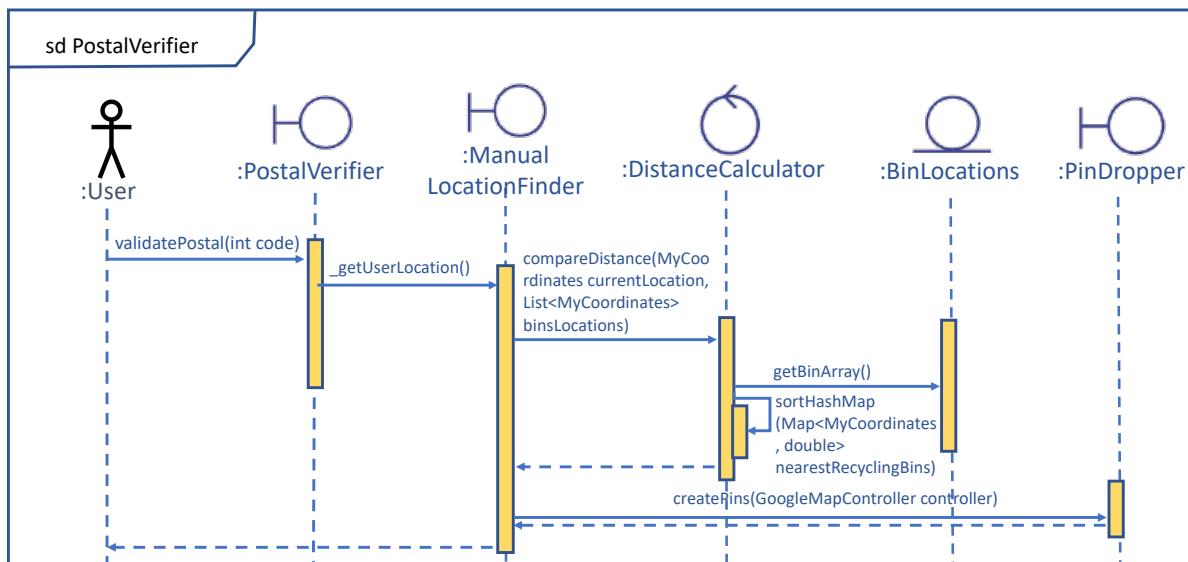
	BinSet	RecyclableSet	PostalCodeSet
System	Yes	Yes	Yes
User	No	No	No
Google Map API	Yes	No	Yes
Data.gov	Yes	No	No

9. Sequence Diagrams

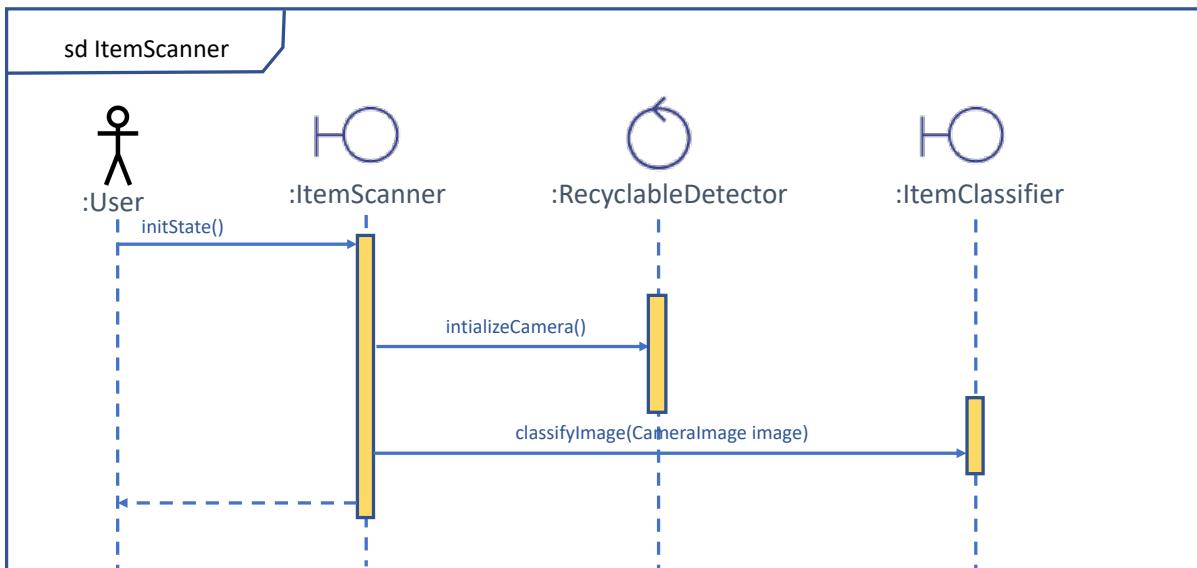
9.1 LocationFinder



9.2 PostalVerifier

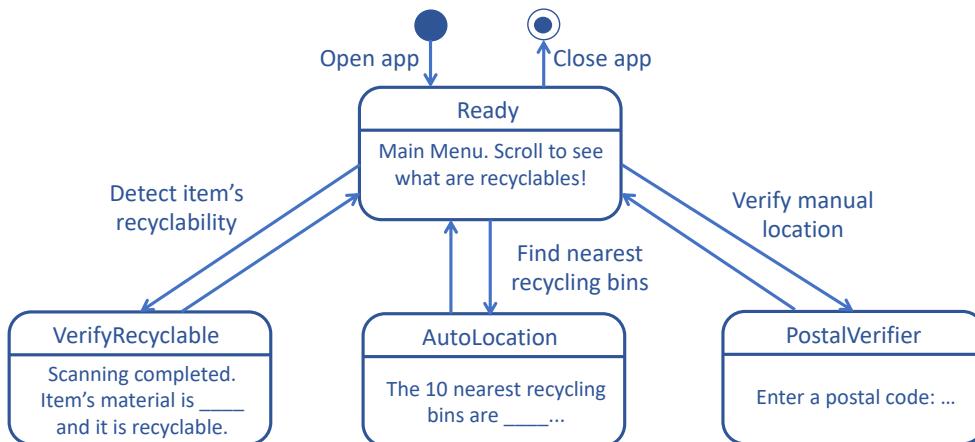


9.3 ItemScanner

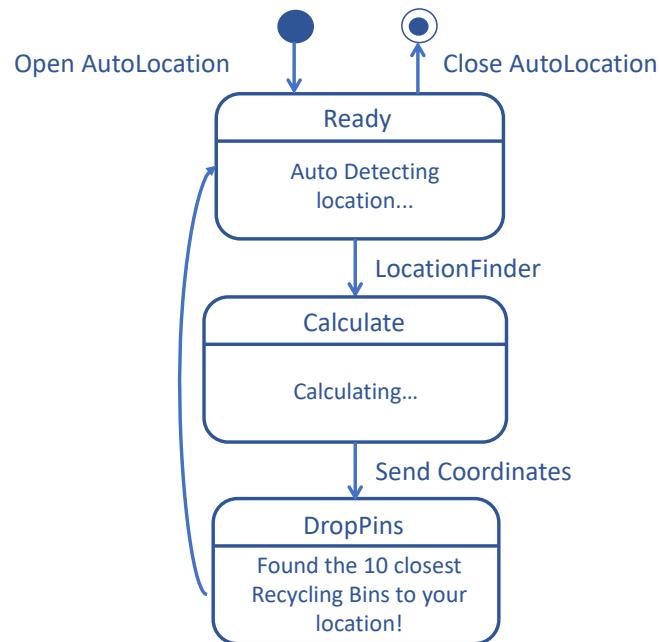


10. Dialog Maps

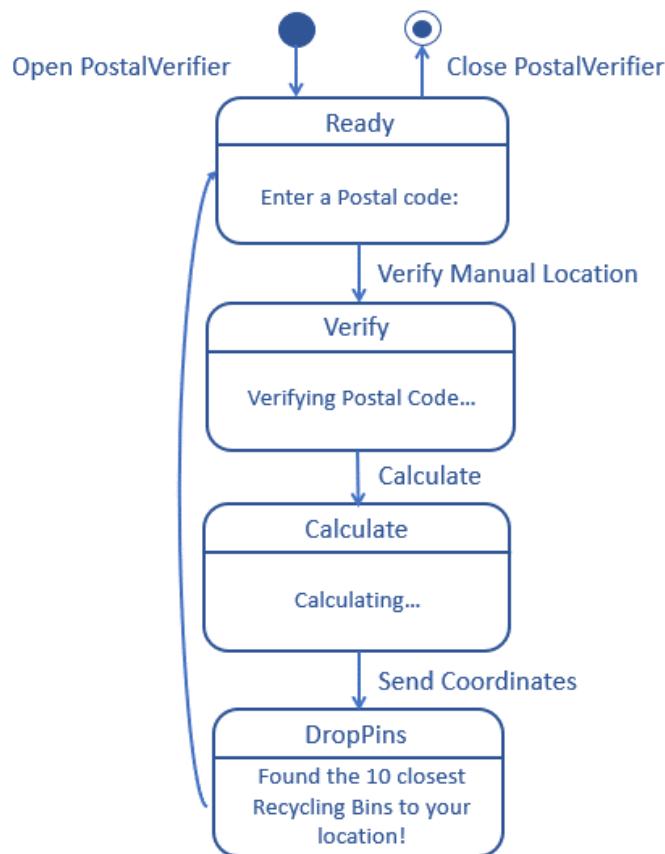
10.1 SG!Recycle



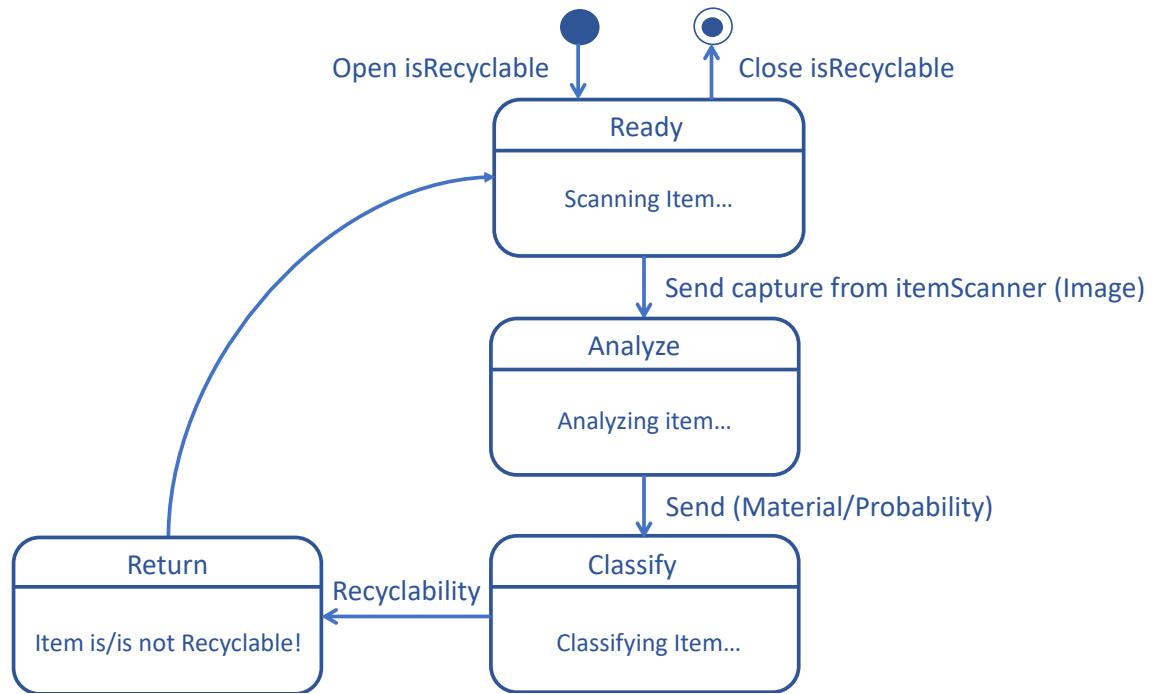
10.2 Auto-Locating Nearest Recycling Bins



10.3 Manual-Locating Nearest Recycling Bins

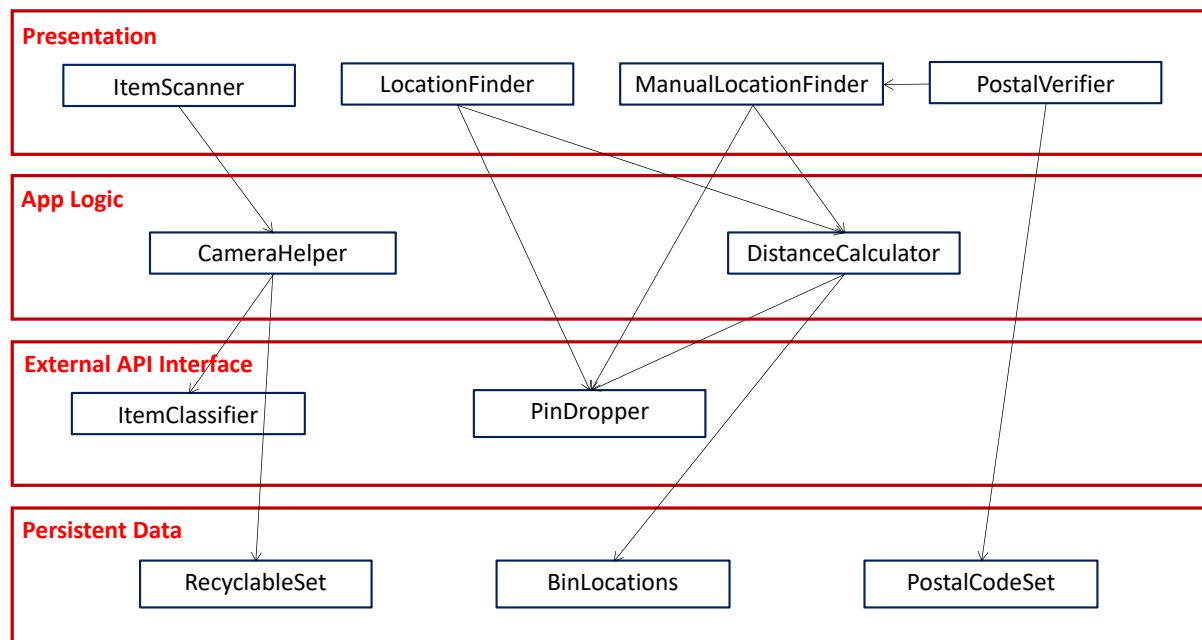


10.4 Classifying User's Item as Recyclable or Not



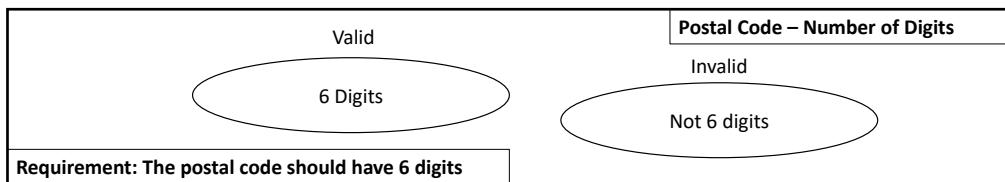
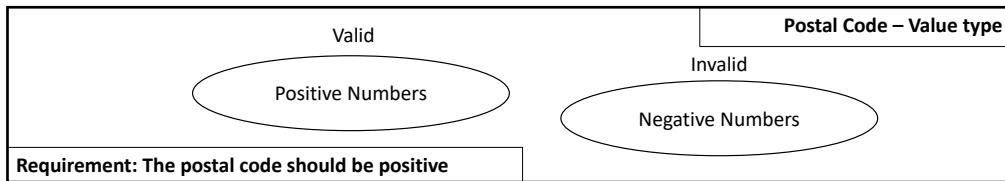
11. System Architecture

We made use of a 4-Tiered Architecture.

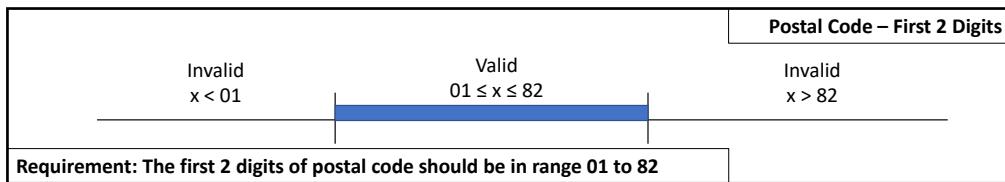


12. Testing

12.1 Equivalence Class Testing for Manual Location Detection



12.2 Boundary Value Testing for Manual Location Detection

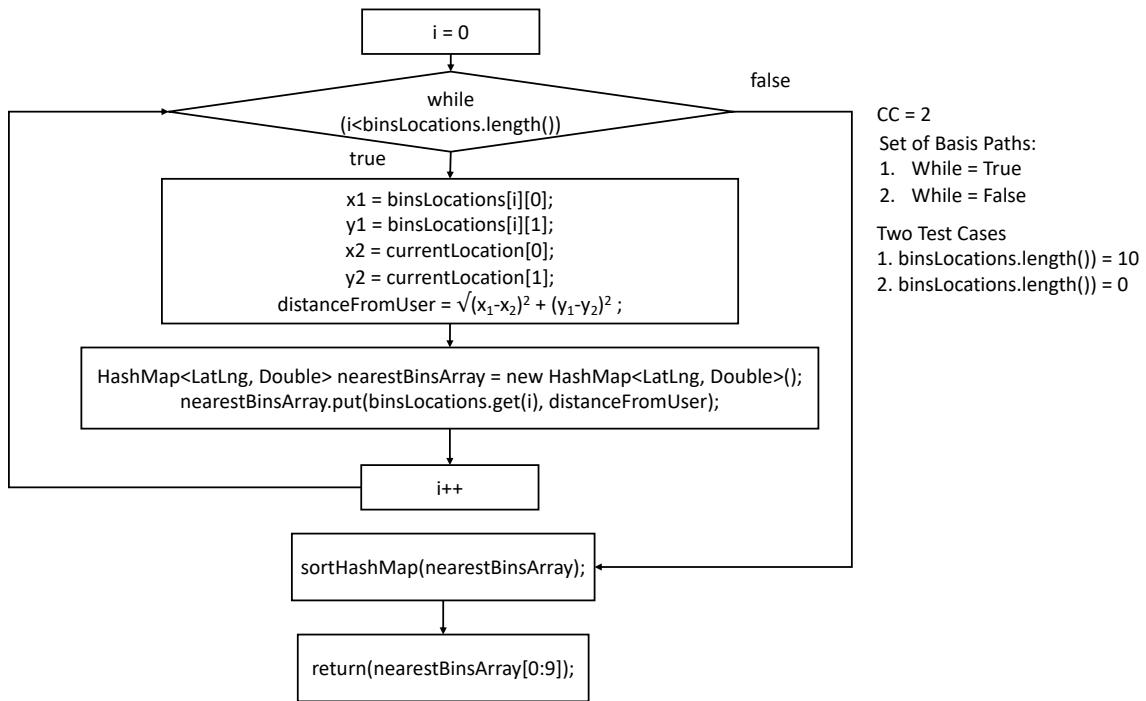


12.3 Combined Test Cases for Manual Location Detection

Manual detection of location

Postal Code	Positive input	Number of Digits	First two Digits	Expected Result	Test Result
2905406	positive	7	29	NaN	NaN
990540	positive	6	99	NaN	NaN
-290540	negative	6	29	NaN	NaN
520840	positive	6	52	1.351100, 103.935600	1.351100, 103.935600
820665	positive	6	82	1.401040, 103.913771	1.401040, 103.913771
730640	positive	6	73	1.438400, 103.802500	1.438400, 103.802500
162004	positive	6	16	1.287500, 103.831000	1.287500, 103.831000

12.4 Control Flow Testing for Distance Calculator



12.5 Control Flow Testing for Item Classifier

