
Software Requirements Specification

for
Hawkerpedia

Version 1.0 approved

Prepared by

Ananya Balehithlu (U2021741A)

Ananya Unnikrishnan (U2021032D)

Charlotte Teo (U2022021G)

Gao Yiping (U2022132D)

Paul Solomon Low Si En (U2022421F)

Roshan Thapa (U2022199F)

Team CRAAPY

8 April 2022

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	2
1.5 References	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	3
2.3 User Classes and Characteristics	4
2.4 Operating Environment	5
2.5 Design and Implementation Constraints	6
2.6 User Documentation	6
2.7 Assumptions and Dependencies	6
3. External Interface Requirements	7
3.1 User Interfaces	7
3.1.1 Home Screen	7
3.1.2 Search Screen	8
3.1.3 Results Screen	9
3.1.4 Information Screen	11
3.1.5 Nearby Carpark Map Screen	12
3.1.6 Hawker Maps Screen	13
3.1.7 Carpark Map	15
3.1.8 Help Screen	17
3.2 Hardware Interfaces	18
3.3 Software Interfaces	18
3.3.1 Amazon Web Services: Relational Database Services (RDS)	18
3.3.2 Backend: Springboot Framework using Java	18
3.3.3 Amazon Web Services: Elastic Beanstalk	18
3.3.4 Data.Gov Live Carpark Lots Availability API	18
3.3.5 Data.Gov Carpark API	18
3.3.6 Hawker Centre Web Scraping Static Data	18
3.4 Communications Interfaces	19
3.4.1 Email Feature	19

4. System Features	20
4.1 Search for Hawker Stalls- CZ001	20
4.2 Apply Filters - CZ002	23
4.3 View list of Hawker Stalls - CZ003	25
4.4 View Specific Hawker Stall Details - CZ004	27
4.5 View Route to Destination - CZ005	29
4.6 View Nearby Carpark List- CZ006	31
4.7 View Specific Carpark Details- CZ007	33
4.8 View Map of all Hawker Centres - CZ008	35
4.9 View Map of all Carparks - CZ009	37
4.10 View Help Page - CZ010	39
5. Other Nonfunctional Requirements	42
5.1 Performance Requirements	42
5.2 Safety Requirements	42
5.3 Security Requirements	42
5.4 Software Quality Attributes	43
5.4.1 Usability	43
5.4.2 Reliability	43
5.4.3 Scalability	43
5.4.4 Supportability	45
5.4.5 Extensibility	45
5.5 Business Rules	45
6. Other Requirements	45
6.1 Reuse Objectives	45
6.2 MVC framework	45
7. Appendix	46
7.1: Appendix A: Data Dictionary	46
7.1.1 Hawker Centres	46
7.1.2 Hawker Centres Closure	46
7.1.3 Carpark Closure	47
7.1.4 Carpark Code	48
7.1.5 Google Maps Routing API	48
7.2 Appendix B: Testing	50
7.2.1 BlackBox Testing - Search Controller Class	50
7.2.2 Whitebox Testing - checkHawkerOpen()	53

7.2.3 Whitebox Testing - updateHistory()	59
7.2.4 Other Testing - Scenario Based Testing	62
7.3 Appendix C: Analysis Models	69
7.3.1 System Architecture Diagram	69
7.3.2 Dialog Map	70
7.3.3 Use Case Diagram	71
7.3.4 Class Diagram	72

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document specifies the software requirements of mobile application Hawkerpedia version 1.0, an iOS and android mobile application. This document will cover the main functionalities of the application, the various user interfaces, as well as the constraints.

Hawkerpedia is a user-friendly mobile application which seeks to improve Singaporeans' hawker food experience by introducing hawker stalls to visit, as well as suggesting directions to get there optimally.

1.2 Document Conventions

The document makes use of the following terms:

Term	Definition
User	Residents of Singapore who want to dine at hawker centres with specific search requirements and are trying to find the shortest route there
Device	Mobile Phone (IOS, Android)
Input	Actions by Users- eg. Search Keywords, Filters Applied
Output	Displays on the screen after input - eg. Search Results of List of Hawker Stalls, Map of Hawker Stalls
Hawker Centre	Contain many stalls selling different cuisines
Hawker Stall	Small booths where individual vendors sell ready-to-eat food

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, and users, to fully understand how the Hawkerpedia mobile application works. All stakeholders are encouraged to read the documentation in sequential order to fully understand the architecture of the Hawkerpedia application.

1.4 Product Scope

From research, 80% of Singaporeans eat out at hawker centres more than once a week. In addition, 9 in 10 strongly agree that hawker centres are a crucial part of Singapore's identity. As such, hawker centres are a significant part of the Singaporean identity and are commonly frequented by Singaporeans.

Thus, our group believes in making the Singaporean identity of hawker stalls accessible for Singaporeans, to enhance our appreciation for Singapore's unique culture and heritage through this app. Therefore, our app is ideal for residents of Singapore who want to dine at hawker centres with specific search requirements and are trying to find the shortest route there. Our app is extremely relevant and useful for Singaporeans in their everyday life.

1.5 References

About Hawkerpedia. (n.d.). Retrieved February 10, 2022, from <https://hawkerpedia.com.sg>.
<https://hawkerpedia.com.sg>

Carpark Availability. (n.d.). Data.Gov.Sg. Retrieved February 1, 2022, from
<https://data.gov.sg/dataset/carpark-availability>

Hawker Centres. (n.d.). Data.Gov.Sg. Retrieved February 8, 2022, from
<https://data.gov.sg/dataset/hawker-centres>

HDB Carpark Information. (n.d.). Data.Gov.Sg. Retrieved February 13, 2022, from
<https://data.gov.sg/dataset/hdb-carpark-information>

2. Overall Description

2.1 Product Perspective

Hawkerpedia is an improved version of current applications that offer information on hawker stalls or carpark parking. By providing a one-stop platform, individuals seeking to patronise hawker centres can find their preferred destination amongst the 1400 stalls present in our extensive database, and how to get there by various modes of transport. By leveraging on filters, users are able to refine their searches, to find hawker stalls that fulfil their demands. In addition, with the asynchronous history storage, users can access the last 10 hawker stalls they viewed on the application, improving overall user experience.

2.2 Product Functions

1. Hawkerpedia allows users to access recently viewed, trending, all time favourite and healthy stalls conveniently.
2. Hawkerpedia allows users to search for hawker stalls by their names or food items.
3. Hawkerpedia allows users to filter the displayed hawker stalls by ratings, distance, neighbourhood and cuisine.
4. Hawkerpedia allows users to view the relevant details such as whether it is currently open, ratings, address, opening hours, and food items of hawker stalls.
5. Hawkerpedia allows users to view the geo-locations of hawker centres, hawker stalls and carparks on a map.
6. Hawkerpedia allows users to view the relevant details such as the real time vacancies available, free parking hours, type of carpark, type of parking system, short-term parking, night parking and gantry height of the carparks within 1km of the hawker centres
7. Hawkerpedia allows users to view the Frequently Asked Questions on how to navigate the app and contact the app developers via email to clarify further doubts.

Please refer to Appendix C to further understand these functionalities.

2.3 User Classes and Characteristics

Hawkerpedia aims to improve individuals' Hawker Centre experience by providing a one-stop platform with information about hawker stalls, its popularity, its availability and how to get there. While any individual who wants to eat at Hawker centres can use Hawkerpedia, there are mainly 2 groups of users with different purposes.

User class	Description
Individuals who are looking for good hawker stalls to eat at	This user class consists of users who want to find good hawker stalls to eat at or to find specific hawker stalls to satisfy their cravings. These users will use most of our functions as they will be looking at the lists of trending, all time favorite and healthy choices to find hawker stalls to explore. They will also be searching for specific hawker stalls or specific food and filtering hawker stalls by cuisine and ratings. After viewing the details of the hawker stalls, they can also view the recently viewed hawker stalls to compare and ultimately choose which hawker stall to eat at. They will then want to view the shortest route there. They are also willing to travel to eat there so they will most likely click the car or public transport route to get there. If they click the route by car, they can also choose which nearby carpark to park their cars at after viewing the real time details of the carparks. These users are the ones expected to use Hawkerpedia the most.
Individuals who want to check for nearby Hawker centres	This user class consists of users who want to find nearby hawker stalls to eat at. These users will also use a lot of our functions. They will mainly view all Hawker centres and choose a hawker centre nearby. They will also use our filter function for distance and neighbourhood frequently. They can also view the hawker stall details to decide which currently open hawker stall to eat at that hawker centre. They will most likely view the shortest walking route to the hawker centre. These users are the second most expected to use Hawkerpedia

2.4 Operating Environment

Backend Development

- MySQL Workbench
- Java : Springboot Framework
 - JDK version 11/17
 - Includes Dependencies: Spring Web, Spring Data JDBC, MySQL Driver
- AWS Relational Database Service
 - Engine: MySQL Community
 - Class: db.t3.micro
- AWS Elastic Beanstalk Service
 - Platform: Corretto 11 running on 64bit Amazon Linux 2/3.2.13
 - Capacity rebalancing: disabled
 - Environment type: single instance
 - Instance types: t2.micro,t2.small
 - Scaling cooldown: 360 seconds
 - Monitoring: CloudWatch Custom Metrics-Environment

External API

- Urban Redevelopment Authority Live Carparks' Lots Availability API
- Data.Gov Hawker Centres API
- Data.Gov Carparks API

Frontend Development

- Javascript: React Native
 - Version: 0.68

Operating Platform

- Development Platform: IOS XCode Simulator and Android Studio
- For Android Studio: The minimum Android SDK version is 28, with SDK version 32 recommended for full functionality.
- Operating System Platform: IOS and Android

2.5 Design and Implementation Constraints

One design limitation we face is the reliance on an external hawker centre website to obtain the ratings for the hawker stalls on our application. Due to the constant changes in hawker centres in Singapore, our developers will be limited in finding ratings for newer stalls as well as less frequented stalls. This will as a result affect the ability to design the hawker stall information page, as we lack the information we aim to present to our user.

2.6 User Documentation

List of User Documentation Components:

1. Demo video to bring the User through on how to use each aspect of the application.
2. User Manual (under User Interfaces in Section 3.1) to list detailed steps and preconditions with illustrated examples to assist the user in using the application, including instructions for troubleshooting problems that crop up when using the application, such as how to deal with different errors and how to obtain help if there is an undocumented problem or an issue they are unable to solve.
3. Online help: Our “Email Feature” will enable users to seek help regarding the usage of the application.
4. Our “FAQ Feature” in the application contains frequently asked questions to allow the user to seek help through answers to commonly asked questions.

2.7 Assumptions and Dependencies

The main underlying assumption is that users using the application have an active internet connection and enabled location services.

Google Maps is also utilised as a third-party application, and therefore we require this service to be installed and working in the users' device for the application to be used accurately.

The application relies on external APIs, such as Data.gov Live Carpark Lots Availability API, thus we require the service to be working for the application to be used accurately.

3. External Interface Requirements

3.1 User Interfaces

3.1.1 Home Screen

The screenshot shows the Hawkerpedia app's home screen. At the top, it displays the time (12:23) and battery status. Below this, the text "Hawkerpedia Presents" and the title "Happy Eating!" are centered. A yellow search bar with the placeholder "Search e.g. Chicken Rice" is positioned below the title. The main content area is divided into two sections: "Recent" and "All Time Favourites". The "Recent" section shows thumbnail images of two stalls: "Feng Sheng Ai..." and "Leong Yeow F...". The "All Time Favourites" section shows thumbnail images of two stalls: "Hock Kee Fried Kway Teow" and "King Avocado". At the bottom, there is a navigation bar with four icons: Home (selected), Maps, Carparks, and Help.

Figure 3.1.1.1: Home Screen

1. Users can scroll the horizontal lists such as Recent, All Time Favourites, Trending and Healthy Choices (Figure 3.1.1.1)
2. Users can click on a hawker stall to navigate to its information screen in Section 3.1.4
 - Recent list will always update and display past 10 hawker stalls clicked throughout the app when user navigates back to home screen (Figure 3.1.1.1)
3. Users can click on Search Bar which would navigate to Search Screen in Section 3.1.2

3.1.2 Search Screen

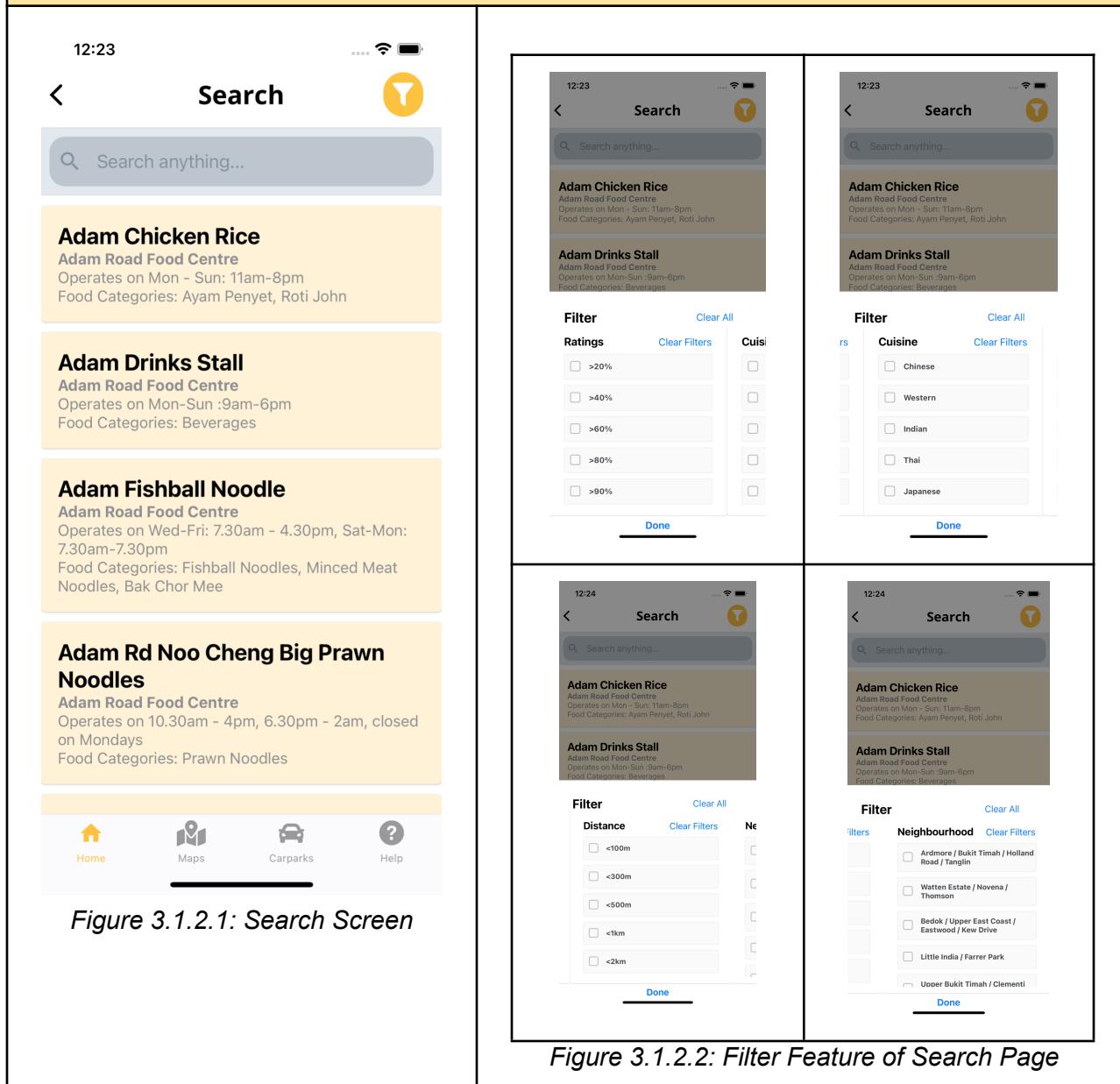


Figure 3.1.2.1: Search Screen

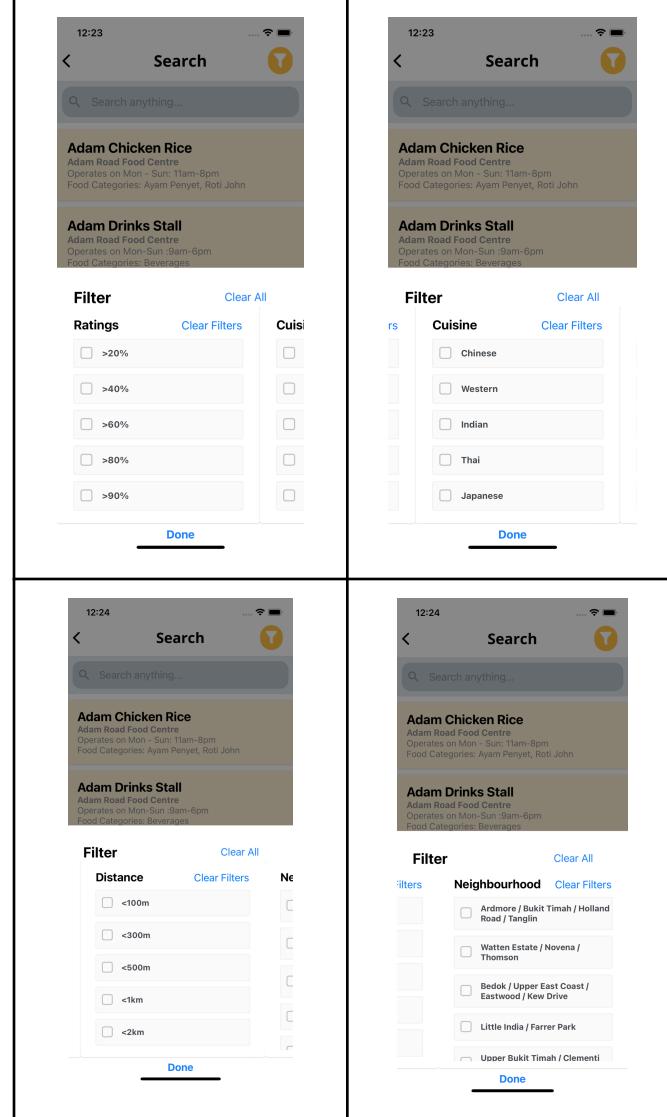


Figure 3.1.2.2: Filter Feature of Search Page

1. Users can key in keywords of their choice in the search bar (Figure 3.1.2.1)
 - Dynamic String matching to hawker stall names is available, enabling user to get a list of suggested hawkers as they type
2. User can click on the suggested hawkers (Figure 3.1.2.1) to navigate to its information screen in Section 3.1.4
3. User can click enter to navigate to the results screen in Section 3.1.3 that matches the User's search keyword (Figure 3.1.3.1)
4. Users can click on the filter button (Figure 3.1.3.1) to open the filter sheet (Figure 3.1.2.2)

5. User can select relevant filters before clicking 'Done' (Figure 3.1.2.2) to navigate to results screen in Section 3.13
6. Users can select 'Clear All' and 'Clear Filters' to deselect all or some of the filter options respectively. (Figure 3.1.2.2)

3.1.3 Results Screen

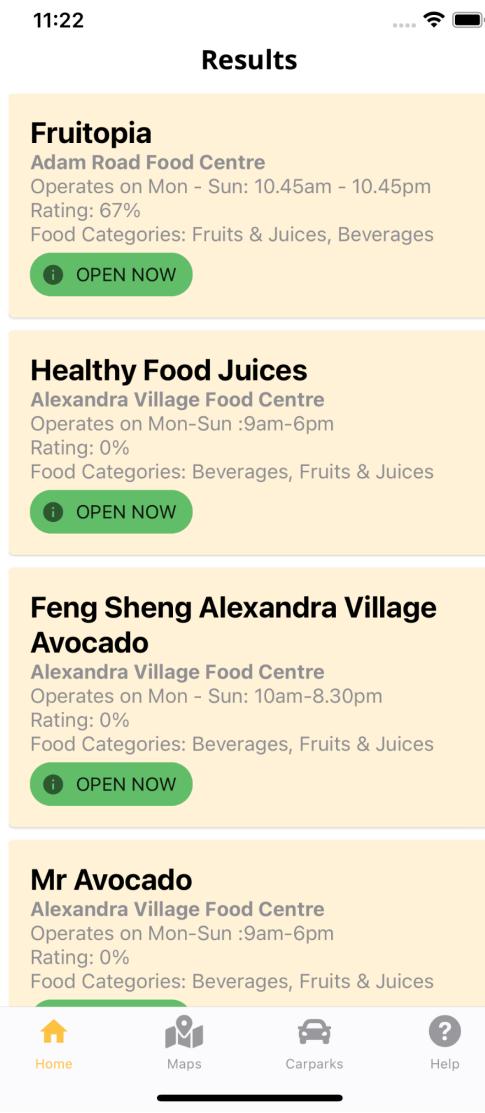


Figure 3.1.3.1: Results Screen

1. Users can scroll vertically to view all the search results (Figure 3.1.3.1)
2. Users can click on the hawker stall of their preference to navigate to its information screen in Section 3.1.4

3. Users can view the summarised information of each hawker stall such as hawker centre name, operation hours, rating, and food categories. The live data of whether the hawker stall is currently open is displayed as well. (Figure 3.1.3.1)

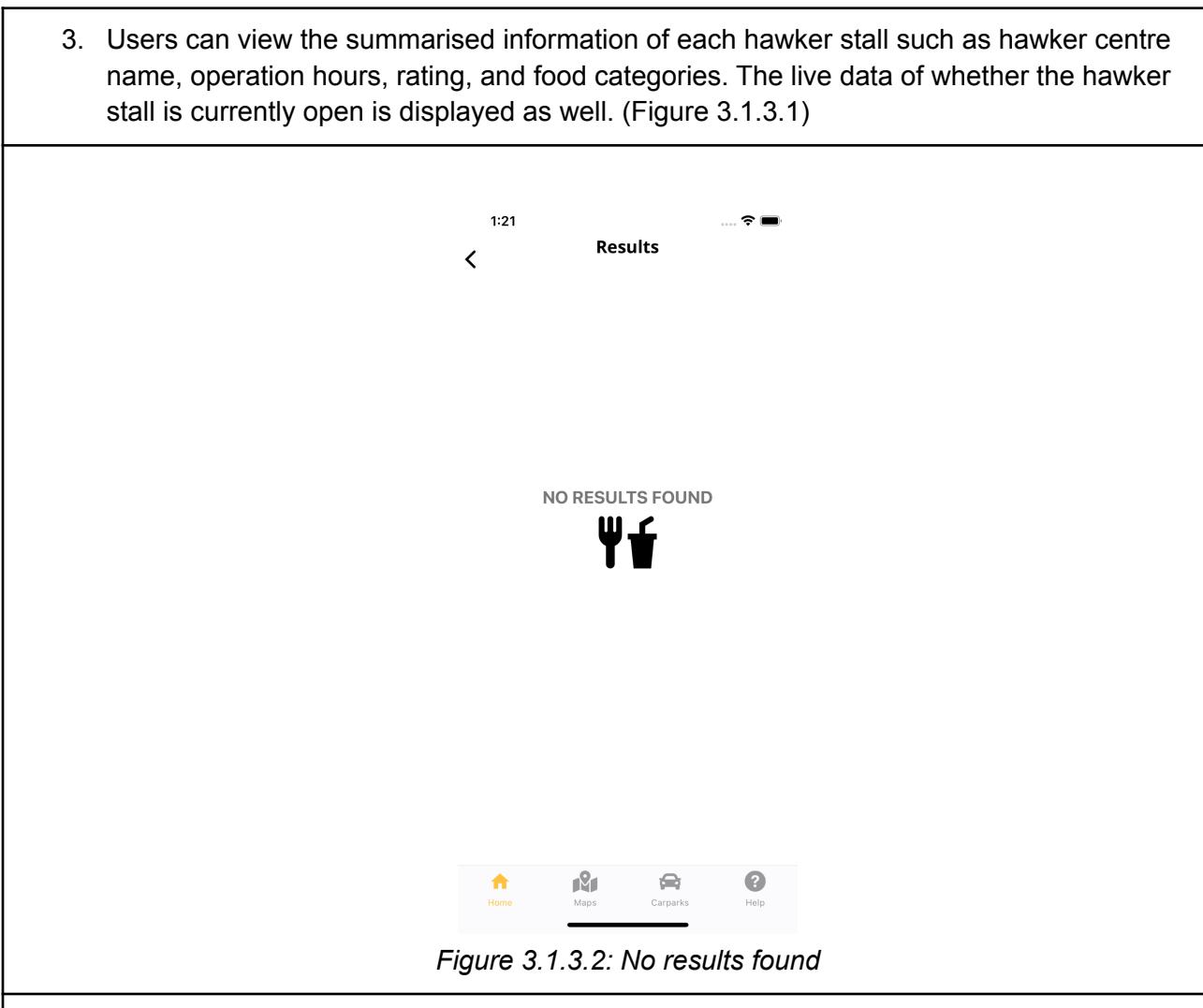


Figure 3.1.3.2: No results found

1. If the user searches a keyword that are not part of a hawker centre name, hawker stall nor food category, the application would display a "No Results Found" Screen, (Figure 3.1.3.2)

3.1.4 Information Screen

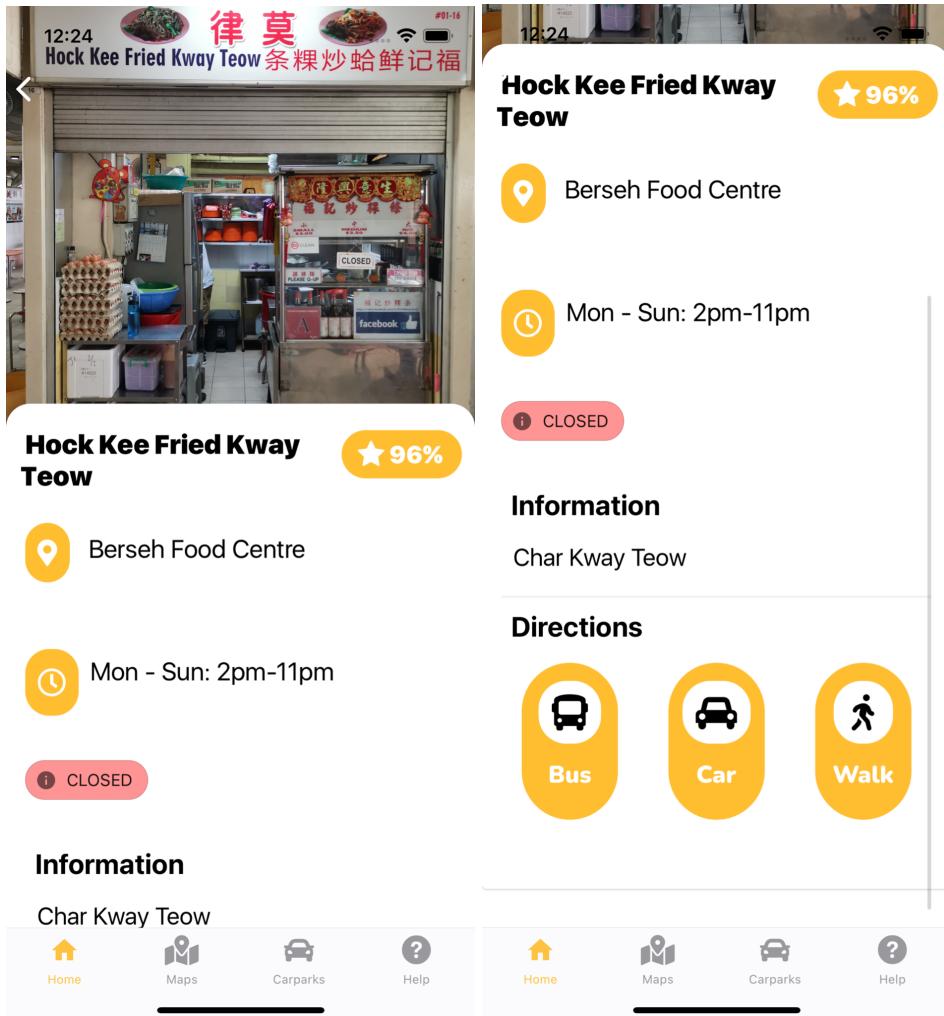
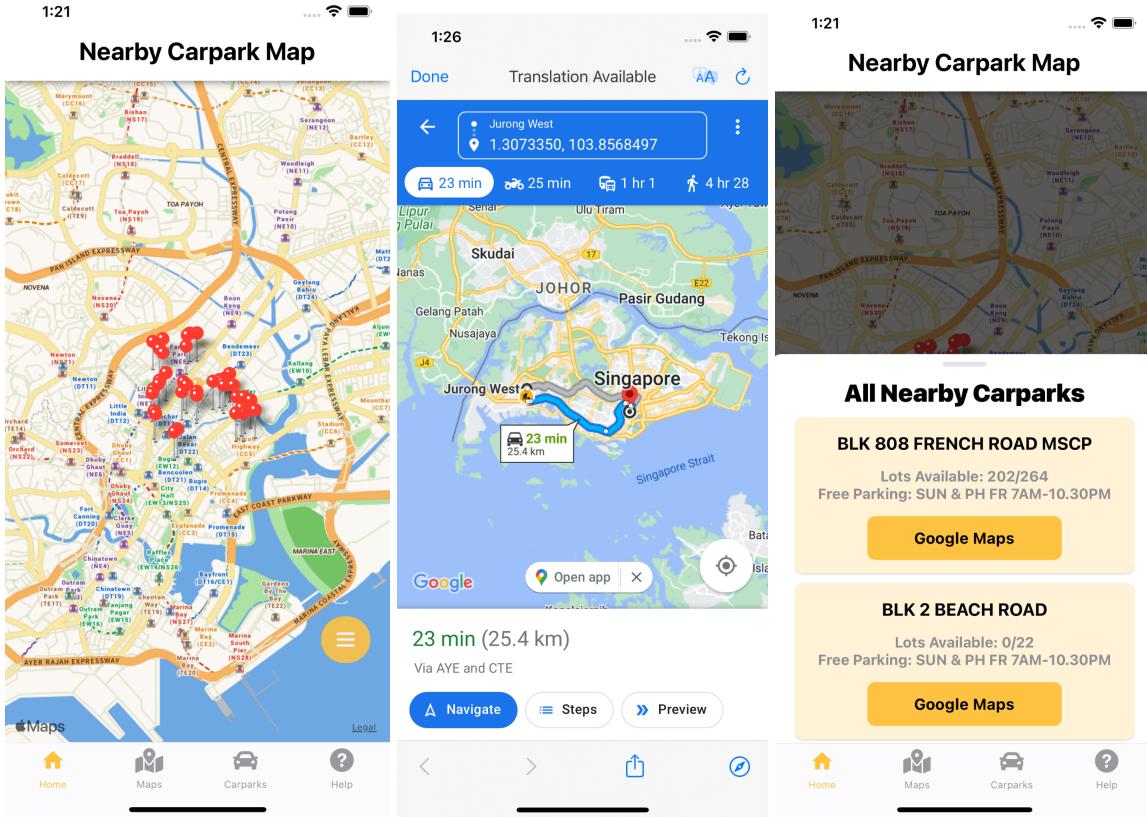


Figure 3.1.4.1: Information Screen

1. Users can view all the detailed information of the hawker stall including directions, ratings and the hawker stall's picture (Figure 3.1.4.1)
2. Users can click on the Bus or Walk button under Directions to open google maps routing from current location
3. Users can click on Car to navigate to Nearest Carpark Maps Screen in Section 3.1.5, to obtain carparks within 1 km from a specific hawker centre.

3.1.5 Nearby Carpark Map Screen



1. Users can interact with the map (Figure 3.1.5.1) by:
 - a. Double Clicking Map to enlarge view
 - b. Moving map with finger
 - c. Click on Marker to see Title
2. Users can double click on marker (Figure 3.1.5.1) to open Alert to open route in google maps (Figure 3.1.5.2)
 - a. Users can also view google maps route (Figure 3.1.5.2) by clicking on the Google Maps button (Figure 3.1.5.3)
3. Users can click on the menu button on the bottom right of Nearby Carpark Map (Figure 3.1.5.1) to display All nearby Carparks (Figure 3.1.5.3).
4. Users can view all nearby carparks along with their summarized information, including Lots Available (Figure 3.1.5.3)
5. Users can click on a carpark (Figure 3.1.5.3) to view additional carpark information (shown in Figure 3.1.7.4)

3.1.6 Hawker Maps Screen

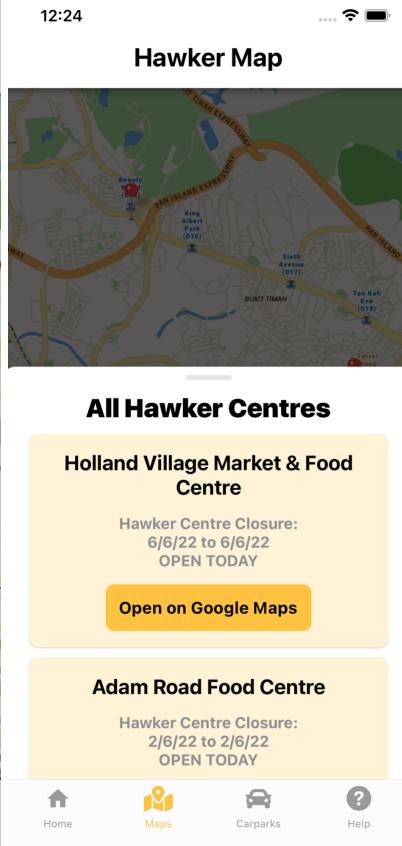
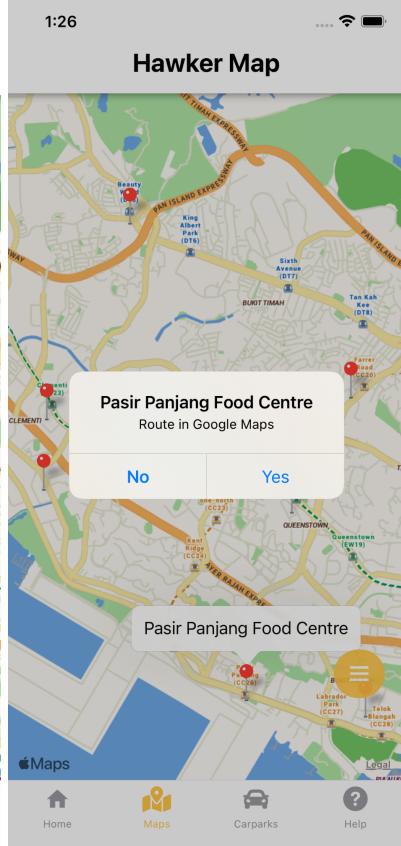
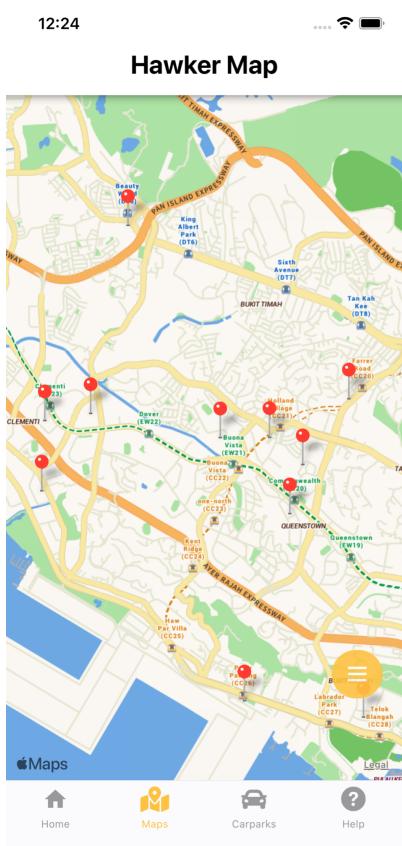


Figure 3.1.6.1:
Hawker Map Screen

Figure 3.1.6.2:
Alert upon clicking marker

Figure 3.1.6.3:
All Hawker Centres

Figure 3.1.6.4:
All Hawker Centres

Figure 3.1.6.5:
Hawker Stalls in Hawker Centre List

Figure 3.1.6.6:
Google Maps Route

1. Users can interact with the map in Figure 3.1.6.1 by:
 - a. Double Clicking Map to enlarge view
 - b. Moving map with finger
 - c. Click on Marker to see Title
2. Users can double click on marker to open Alert (Figure 3.1.6.2) to open route in google maps (Figure 3.1.6.6)
 - a. Users can also view route in google maps (Figure 3.1.6.6) by clicking on "Open on Google Maps" (Figure 3.1.6.4)
3. Users can drag up the All Hawker Centres tab from (Figure 3.1.6.3) to (Figure 3.1.6.4)
4. Users can view the summarized information of each hawker stall such as hawker centre closure dates. The live data of whether the hawker stall is currently open is displayed as well. (Figure 3.1.6.4)
5. Users can scroll list to view hawker centres and their hawker closure information (Figure 3.1.6.4)

6. Users can click on a hawker centre, to enter results screen of all hawker stalls within the hawker centre (Figure 3.1.6.5)

3.1.7 Carpark Map

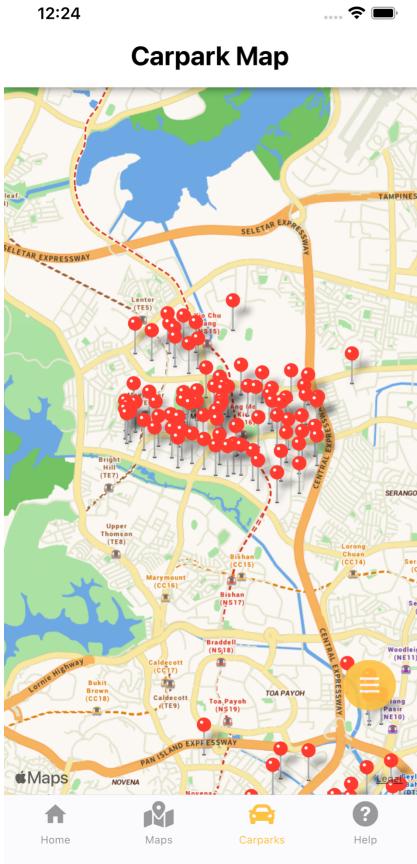


Figure 3.1.7.1:
Carpark Map Screen

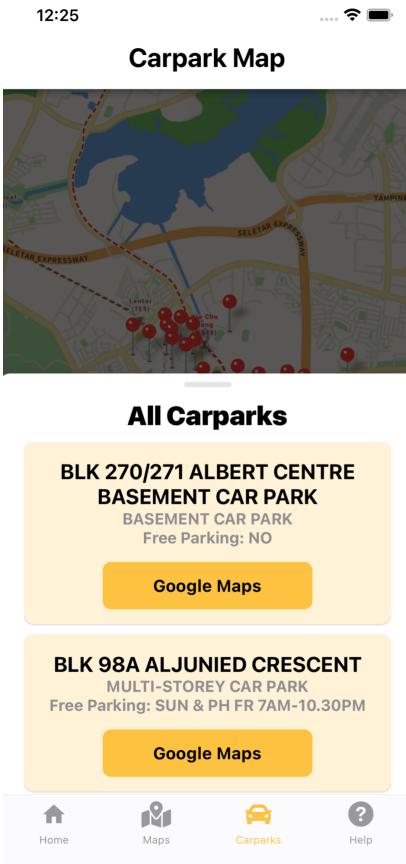


Figure 3.1.7.2:
All Carparks

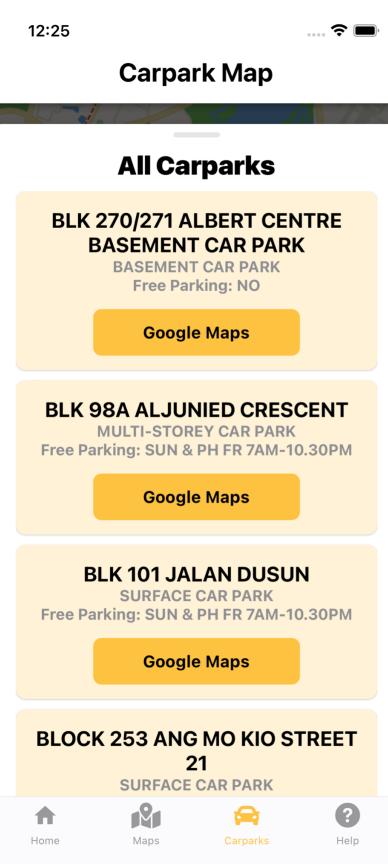


Figure 3.1.7.3:
All Carparks

1. Users can interact with the map (Figure 3.1.7.1) by:
 - 1.1. Double Clicking Map to enlarge view
 - 1.2. Moving map with finger
 - 1.3. Click on Marker to see Title
2. Users can double click on marker to open Alert (Figure 3.1.7.1) to open route in google maps
 - 2.1. Users can also open route to google maps by pressing "Google maps" in All

Carparks menu (Figure 3.1.7.2)

3. Users can open all carparks menu by pressing the menu on the bottom right of the Carpark Map Screen (Figure 3.1.7.1)
4. Users can scroll list to view carparks and their type and free parking timings. (Figure 3.1.7.3)
5. Users can click on a carpark (Figure 3.1.7.3) to enter carpark information screen (shown in Figure 3.1.7.4) to view additional details of carpark

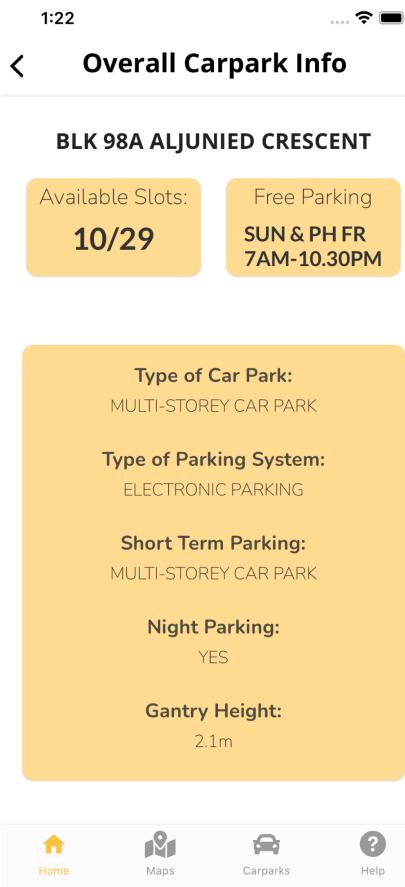


Figure 3.1.7.4: Additional Carpark Information

1. Users can view relevant information about carpark, such as real-time lots availability and type of parking available. (Figure 3.1.7.4)

3.1.8 Help Screen

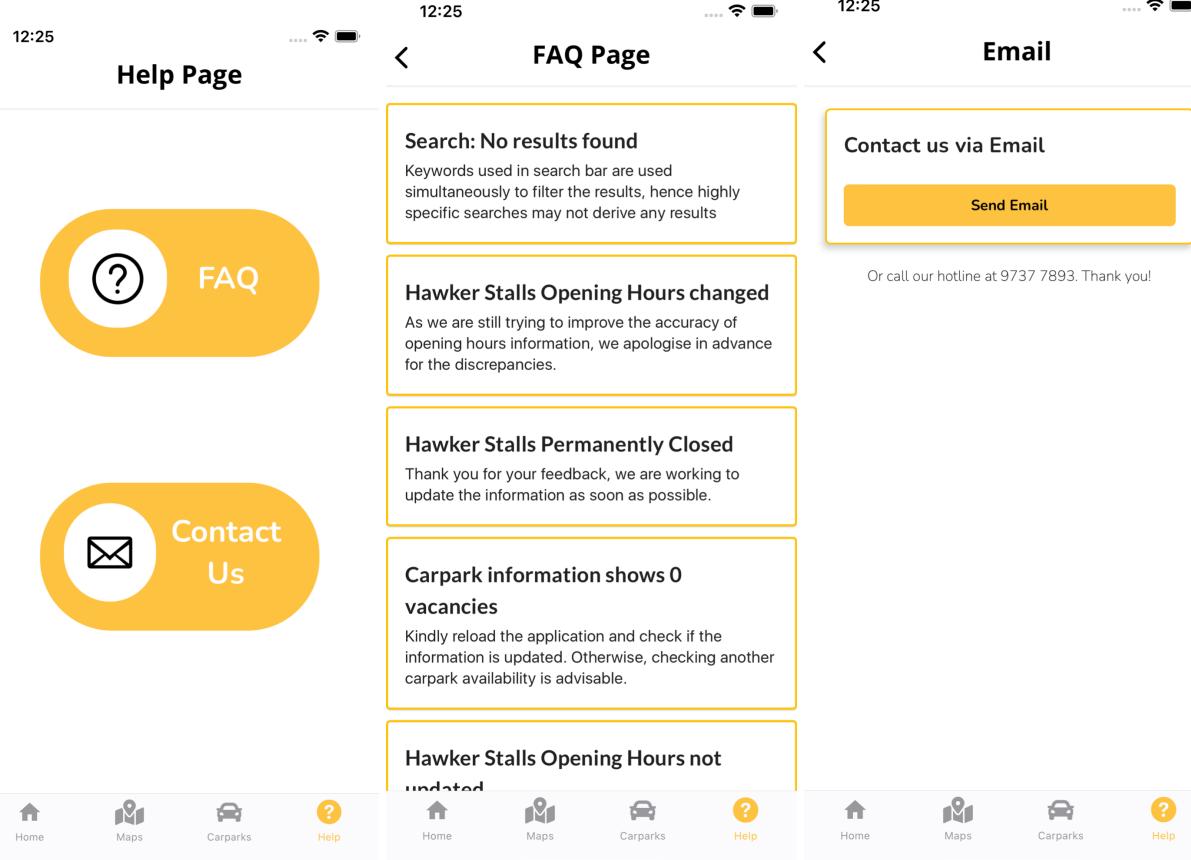


Figure 3.1.8.1:
Help Page Screen

Figure 3.1.8.2:
FAQ Page Screen

Figure 3.1.8.3:
Email Screen

1. Users can click on the FAQ button (Figure 3.1.8.1), which would then navigate to FAQ Screen (Figure 3.1.8.2) with a list of commonly asked questions.
2. Users can click on the Contact Us button (Figure 3.1.8.1) which would bring them to the Email Screen. (Figure 3.1.8.3) Clicking on "Send Email" would open the Email Application on User's phone, and fill both:
 - a. Subject Title: "Query to Hawkerpedia"
 - b. Recipient: contactus@hawkerpedia.com in the email draft.

Please refer to Section 7.2.4 Appendix B for testing of User Interfaces.

3.2 Hardware Interfaces

The application does not require any hardware to run on.

3.3 Software Interfaces

3.3.1 Amazon Web Services: Relational Database Services (RDS)

The Hawkerpedia application leverages on **MySQL workbench** to store data under AWS RDS. This relational database service by Amazon is setup on cloud which eases frontend deployment. As RDS is built upon MySQL, the database can easily handle large amounts of data across multiple tables, in this case, our hawker stalls and carparks.

3.3.2 Backend: Springboot Framework using Java

Springboot is a framework that enables you to decouple your program . It makes switching components or implementations easier by using dependency-injection. By connecting Springboot to our MySQL database, we are able to carry out SQL queries and get requests in real time and map it to a link. Additionally, Springboot is also used to connect our database to the Data.gov API, to obtain real time carpark lot availability.

3.3.3 Amazon Web Services: Elastic Beanstalk

The Elastic Beanstalk Service enables the entire Springboot backend to be built on cloud upon uploading its .JAR file. This service allows for easy access to queried data for frontend development.

3.3.4 Data.Gov Live Carpark Lots Availability API

The live carpark data enables us to obtain real-time updates on the lots available for all the carparks in Singapore by providing the carparks' code and their respective lots available.

3.3.5 Data.Gov Carpark API

As the Live Carpark data does not contain sufficient information on carparks, such as address and types of parking, we leveraged on the data.gov carpark API and joined it with 3.3.5 Data.gov API in the backend.

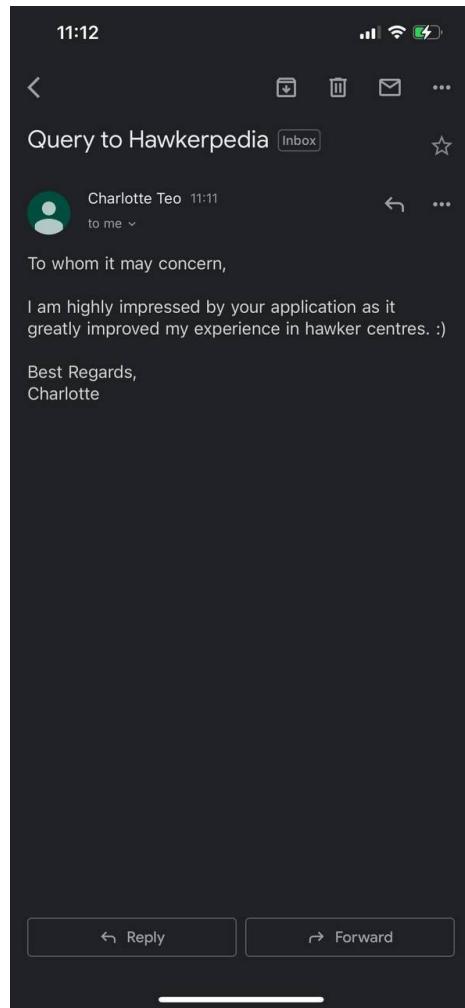
3.3.6 Hawker Centre Web Scraping Static Data

As Hawkerpedia requires a large amount of information on hawker stalls, such as their operation hours, pictures and food categories, we could not find any datasets online that fulfilled all these criterias. Thus, we applied web scraping on food review websites, to obtain the necessary information. Thereafter, we uploaded the data into the MySQL database.

3.4 Communications Interfaces

3.4.1 Email Feature

Upon clicking on “Contact Us” in the Email Screen, the Email application would open on the user’s phone. An email draft would be automatically generated for the user with the Email Subject Title as “Query to Hawkerpedia” and the recipient being Hawkerpedia’s Email.



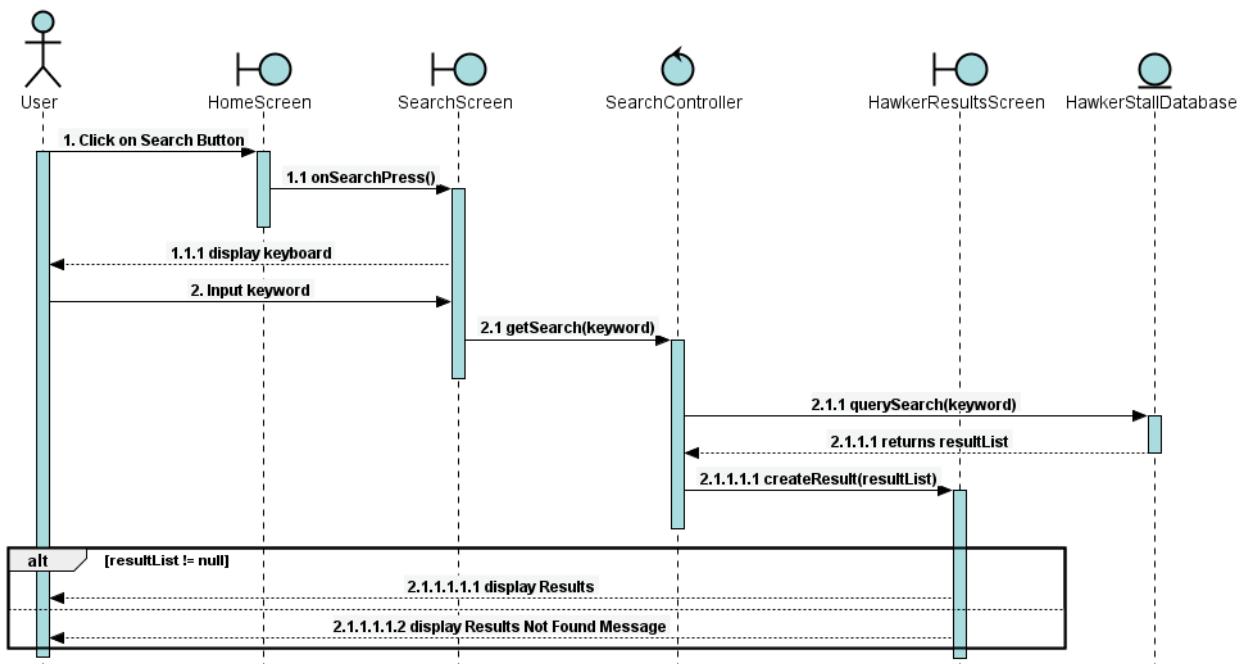
4. System Features

4.1 Search for Hawker Stalls- CZ001

Use Case ID:	CZ001		
Use Case Name:	Search for Hawker Stalls		
Created By:	Yiping	Last Updated By:	Yiping
Date Created:	31 Jan 2022	Date Last Updated:	3 Apr 2022

Actor:	User
Description:	User selects the Search button to search for hawker stalls by inputting their desired food item or the name of the hawker stall/centre.
Preconditions:	<ol style="list-style-type: none"> 1. The user must have an active internet connection. 2. The user needs to press the Search button on the Home Page
Postconditions:	<ol style="list-style-type: none"> 1. The user will have a list of search results that they can choose from or further filter in Use Case CZ002
Priority:	High
Frequency of Use:	High. Search function is essential when the user wants to quickly find a specific hawker stall by name or all the hawker stalls in the specific hawker centre or that provide the food item they want.
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks on the search button on the Home page. 2. The app will display the search page 3. The user will press on the search bar to display the keyboard 4. User will key in their desired food item, hawker centre or hawker stall name and press search 5. The app will display a list of Hawker stalls that match the input from the user leading to Use Case CZ003

	6. User may extend to Use Case CZ002 to apply filters to their search results
Alternative Flows:	AF-S1: If there exists no match to the search keyword the user has entered: 1. App will display "Results not found" with an icon 2. Return to Step 3
Exceptions:	NIL
Includes:	1. Use Case CZ003 View List of Hawker Stalls
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

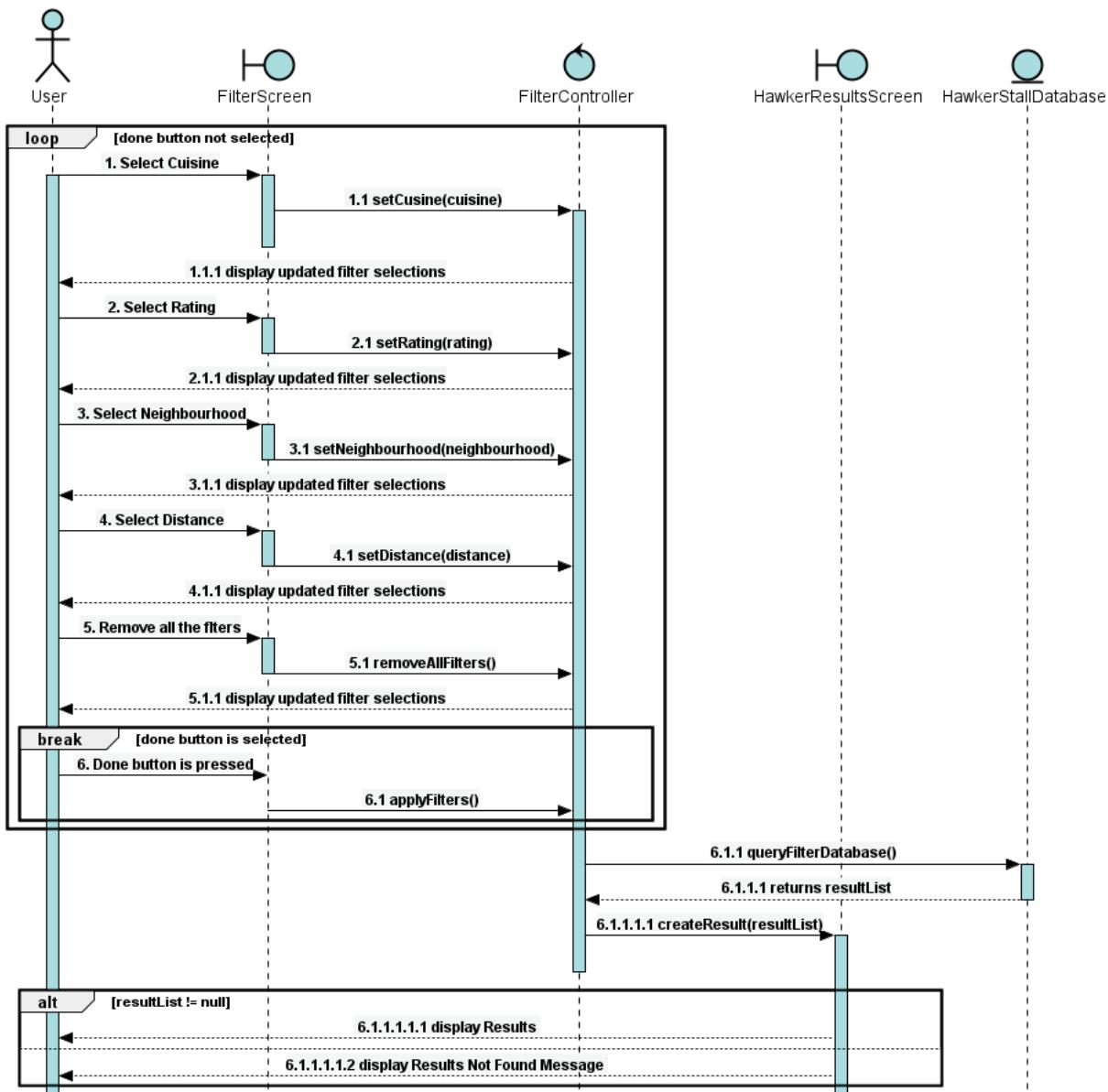


4.2 Apply Filters - CZ002

Use Case ID:	CZ002		
Use Case Name:	Apply Filters		
Created By:	Paul, Charlotte	Last Updated By:	Charlotte
Date Created:	23 Jan 2022	Date Last Updated:	5 Apr 2022

Actor:	User
Description:	User applies filter conditions to generate a list of hawker stalls that fit the user's request
Preconditions:	<ol style="list-style-type: none"> 1. The user must have location services turned on and location permissions granted. 2. The user must have an active internet connection. 3. The user needs to press the filter button on the app
Postconditions:	<ol style="list-style-type: none"> 1. The user will have a list of hawker stalls that match the filter conditions leading to Use Case CZ003
Priority:	High
Frequency of Use:	High. Filter function is quintessential in optimising the results shown to users
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks the filter button at the search page. 2. The app will display 4 sets of filter checkboxes for the user to select filters: Ratings, Distance, Cuisine and Neighbourhood. 3. User will select the desired filter option based on their requirements. 4. User will press 'Done' to search based on the selected filter fields. 5. The app will display a resulting list of hawker stalls that matches the filter requirements as mentioned in Use Case CZ003.
Alternative Flows:	<p>AF-S1: If user wants to reset filters</p> <ol style="list-style-type: none"> 1. User will select the 'Clear All' button

	2. App will remove all previously applied filters 3. User will return to step 3
Exceptions:	NIL
Includes:	1. Use Case CZ003 View List of Hawker Stalls
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL



4.3 View list of Hawker Stalls - CZ003

Use Case ID:	CZ003		
Use Case Name:	View list of Hawker Stalls		
Created By:	Roshan	Last Updated By:	Roshan
Date Created:	31 Jan 2022	Date Last Updated:	20 Mar 2022

Actor:	User
Description:	User views a list of hawker stalls after Use Case CZ002 or Use Case CZ001
Preconditions:	<ol style="list-style-type: none"> 1. The user's device needs to be currently connected to the internet 2. The user's device needs to enable location services under their privacy settings 3. The user needs to execute Use Case CZ001 or Use Case CZ002
Postconditions:	<ol style="list-style-type: none"> 1. The user can now see the list of the relevant hawker stalls. 2. The list of hawker stalls will display the hawker stall name, hawker centre name, rating, operation hours and the food items it sells. 3. The user may extend to Use Case CZ004 to view full details of a specific stall.
Priority:	High. Viewing the list of hawker stalls is very critical in helping users select specific hawker stalls to patronise
Frequency of Use:	High. Users are likely to frequently view the list of hawker stalls every time after successfully executing Use Case CZ001 or Use Case CZ002
Flow of Events:	<ol style="list-style-type: none"> 1. The user must successfully carry out Use Case CZ001 or CZ002 2. The user can view the list of hawker stalls 3. The user may extend to Use Case CZ004 to view full details of a specific stall.

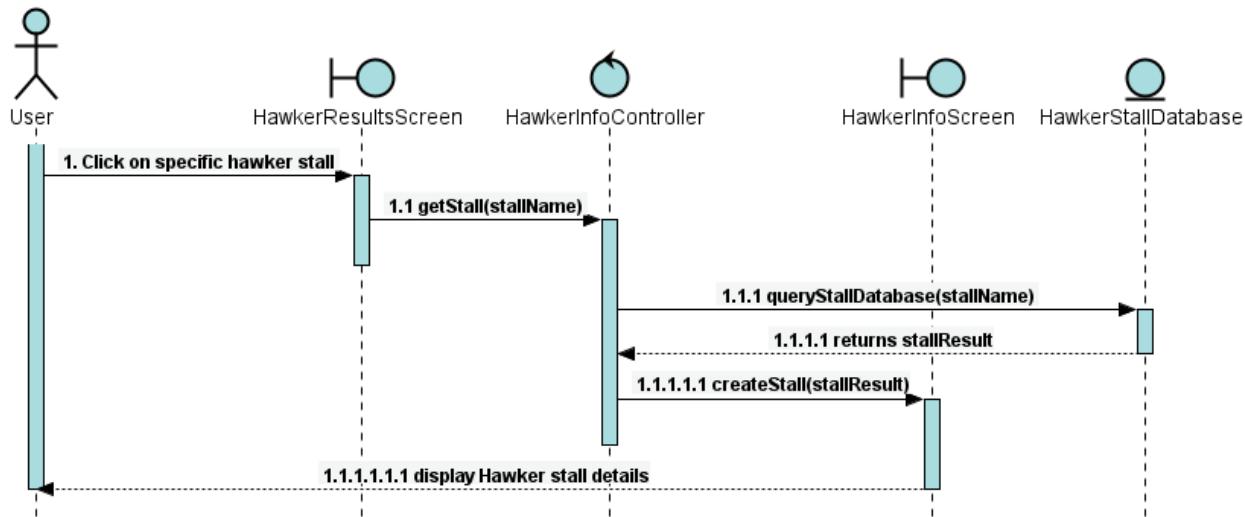
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

4.4 View Specific Hawker Stall Details - CZ004

Use Case ID:	CZ004		
Use Case Name:	View Specific Hawker Stall Details		
Created By:	Roshan	Last Updated By:	Roshan
Date Created:	22 Jan 2022	Date Last Updated:	3 Apr 2022

Actor:	User
Description:	Users select a specific hawker stall from list of hawker stalls to get the full details
Preconditions:	<ol style="list-style-type: none"> 1. The user's device needs to be currently connected to the internet 2. The user's device needs to enable location services under their privacy settings 3. The user needs to select a specific hawker stall after Use Case CZ003
Postconditions:	<ol style="list-style-type: none"> 1. The user can now see the full details of a specific hawker stall such location, ratings etc
Priority:	High. Viewing the full details of a specific hawker stall is very critical in helping users choose where to eat, which is the primary purpose of this app.
Frequency of Use:	High. Users are likely to frequently view the full details of a specific hawker stall.
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks a specific hawker stall from the list of hawker stalls after Use Case CZ003 2. The user can view a page containing full details of a specific hawker stall such as cuisine, location, ratings etc. 3. The user may extend to Use Case CZ005 to view the shortest route to hawker stall.
Alternative Flows:	NIL
Exceptions:	NIL

Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL



4.5 View Route to Destination - CZ005

Use Case ID:	CZ005		
Use Case Name:	View Route to Destination		
Created By:	Ananya U	Last Updated By:	Ananya U
Date Created:	22 Jan 2022	Date Last Updated:	23 Mar 2022

Actor:	User
Description:	The User chooses a mode of transport to view the shortest route to the hawker stall.
Preconditions:	<ol style="list-style-type: none"> 1. The user must have location services turned on and location permissions granted. 2. The user must have an active internet connection.
Postconditions:	<ol style="list-style-type: none"> 1. User is able to see the fastest route on Google Maps from their current location to the selected hawker stall after choosing the mode of transport as “Walk” or “Bus”. 2. User may extend to Use Case CZ006 to view a list of carparks with vacancies near the hawker stall after choosing “Car” as the mode of transport.
Priority:	Medium. Viewing the hawker stall on a map, both in terms of a spatial view and distance, will allow for the user to make a more informed decision when choosing the desired hawker stall.
Frequency of Use:	High. The user will access the map every time they wish to view the stall location and route.
Flow of Events:	<ol style="list-style-type: none"> 1. The user can choose a mode of transport i.e. walking, public transport or car. 2. The user can view the shortest route from the current location by public transport or walking on Google Maps. 3. The user can extend to Use Case CZ006 after choosing car as mode of transport.

Alternative Flows:	AF-S1: If location service is turned off or location service permission is not granted 1. The application will prompt the user for location services to be turned on and request for location services permission.
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

4.6 View Nearby Carpark List- CZ006

Use Case ID:	CZ006		
Use Case Name:	View nearby carparks with vacancies for parking		
Created By:	Ananya B	Last Updated By:	Ananya B
Date Created:	24 Jan 2022	Date Last Updated:	2 Apr 2022

Actor:	User
Description:	The user can view a list of carparks near the selected hawker stall.
Preconditions:	<ol style="list-style-type: none"> 1. The user must have selected a hawker stall 2. The user must have selected “car” under “directions” as mode of transport from Use Case CZ005 3. The user must have an active internet connection 4. The user must have location services turned on and location permissions granted
Postconditions:	<ol style="list-style-type: none"> 1. The user is able to view all the nearby carparks within 1km as markers on a map view. 2. Upon clicking the bottom right button, he user is able to view the list of nearby carparks with the number of available slots and if there is free parking. 3. The user can click the “Google Maps” button to be redirected to google maps to view the shortest route.
Priority:	High - Viewing the carpark and its available slots from the hawker centre will enable users to make informed decisions while choosing the carpark and final mode of transport
Frequency of Use:	High. The user will access the list of nearby carparks every time their mode of transport is by car.
Flow of Events:	<ol style="list-style-type: none"> 1. The app will display a map with the nearby carparks within 1km as markers. 2. When the user clicks the bottom right button, the app will display a list of nearby carparks. 3. Users have 2 options from here

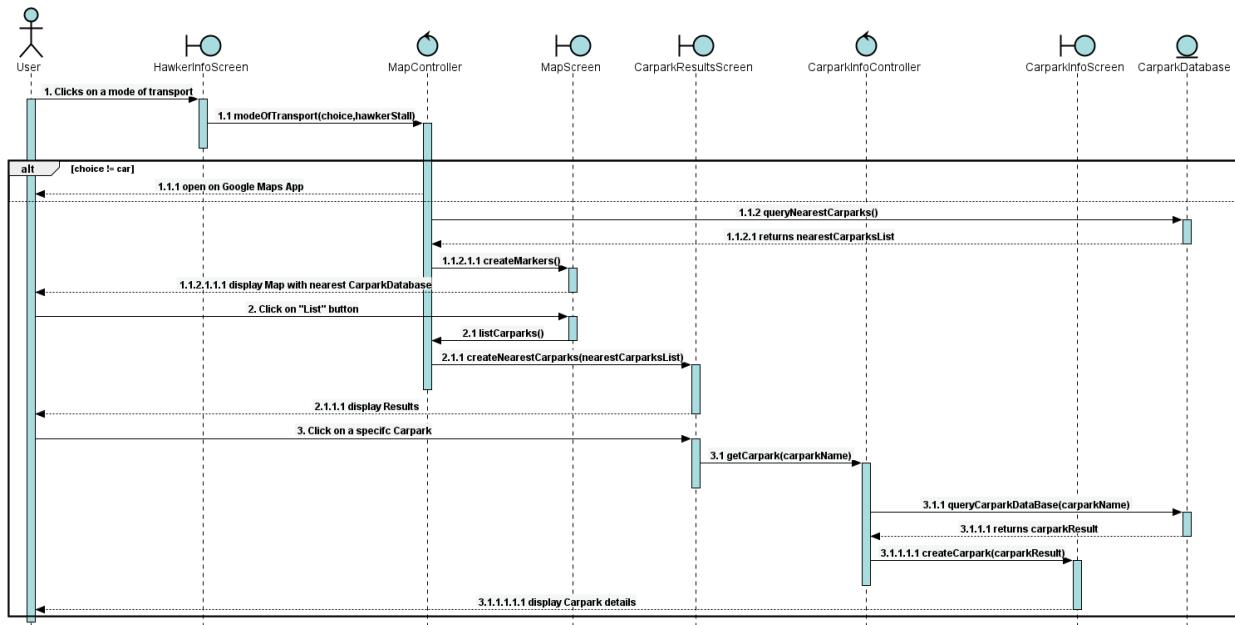
	<ul style="list-style-type: none"> a. User can click “Google Maps” to be redirected to google maps to view shortest route. b. User can click on the carpark name to be redirected to Use Case CZ007. <ul style="list-style-type: none"> i. User can once again click “Google Maps” to be redirected to google maps to view the shortest route.
Alternative Flows:	<p>AF-S1: If location service is turned off or location service permission is not granted</p> <ol style="list-style-type: none"> 1. The application will prompt the user for location services to be turned on and request for location services permission.
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

4.7 View Specific Carpark Details- CZ007

Use Case ID:	CZ007		
Use Case Name:	View Specific Carpark Details		
Created By:	Ananya B	Last Updated By:	Ananya B
Date Created:	22 Jan 2022	Date Last Updated:	18 Mar 2022

Actor:	User
Description:	The user selects a specific carpark from the list to view the full details of the carpark.
Preconditions:	<ol style="list-style-type: none"> 1. The user's device needs to be currently connected to the internet 2. The user's device needs to enable location services under their privacy settings 3. The user must have clicked on the carpark name.
Postconditions:	<ol style="list-style-type: none"> 1. User is able to view the available slots, the type of carpark, parking system, the gantry height, whether the parking is free/night-time can also be viewed on this page.
Priority:	Medium. Viewing the full details of a specific carpark will help users choose where to park.
Frequency of Use:	Medium. Users may want to view the full details of a carpark.
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks a specific carpark from the list of carparks after Use Case CZ006. 2. The user can view a page containing full details of a specific carpark such as type of carpark, parking system, gantry height etc.
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL

Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

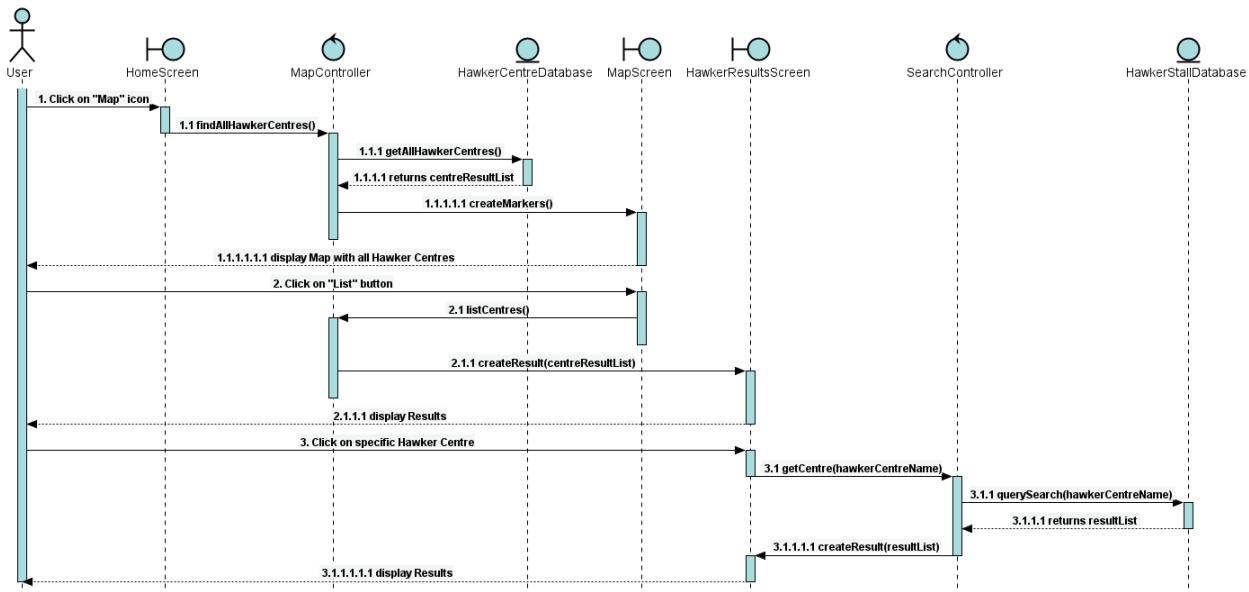


4.8 View Map of all Hawker Centres - CZ008

Use Case ID:	CZ008		
Use Case Name:	View Map of all Hawker Centres		
Created By:	Ananya U	Last Updated By:	Ananya U
Date Created:	31 Jan 2022	Date Last Updated:	19 Mar 2022

Actor:	User
Description:	User selects the map icon to view all of the hawker centres as markers on a map as well as in a pop-up list.
Preconditions:	<ol style="list-style-type: none"> 1. The user must have location services turned on and location permissions granted. 2. The user must have an active internet connection. 3. The user must press the map icon next to the home page icon to view all hawker centres on a map.
Postconditions:	<ol style="list-style-type: none"> 1. Users are able to spatially view all the hawker centres as markers on a map. 2. Clicking on the marker will lead the user to google maps. 3. The user can zoom in and out of the map by pinching. 4. The user is able to click on the “List” button to view a list of hawker centres. <ul style="list-style-type: none"> a. The user can view basic information about the hawker centre such as its name, hawker stall closure and whether it's open on that day. b. The user can click on “Google Maps” button to open on google maps c. The user can click on the hawker centre name to view a list of all hawker stalls in that hawker centre . d. The user may extend to Use Case CZ004 after selecting a specific hawker stall form the list of hawker stalls

Priority:	Medium. Viewing the hawker centres on a map will allow for the user to make a more informed decision when choosing the desired hawker stall.
Frequency of Use:	Medium. The user will access the map every time they wish to view all of the hawker centre locations.
Flow of Events:	<ol style="list-style-type: none"> 1. The user can view all hawker centres in Singapore as markers on the map after selecting the map icon next to the home page icon. 2. The user can click on the “List” button to view all hawker centres in a list. 3. The user may click on a specific hawker centre to view a list of hawker stalls in the selected hawker centre. 4. The user may extend to Use Case CZ004 after selecting a specific hawker stall from the list.
Alternative Flows:	<p>AF-S1: If location service is turned off or location service permission is not granted</p> <ol style="list-style-type: none"> 1. The application will prompt the user for location services to be turned on and request for location services permission.
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

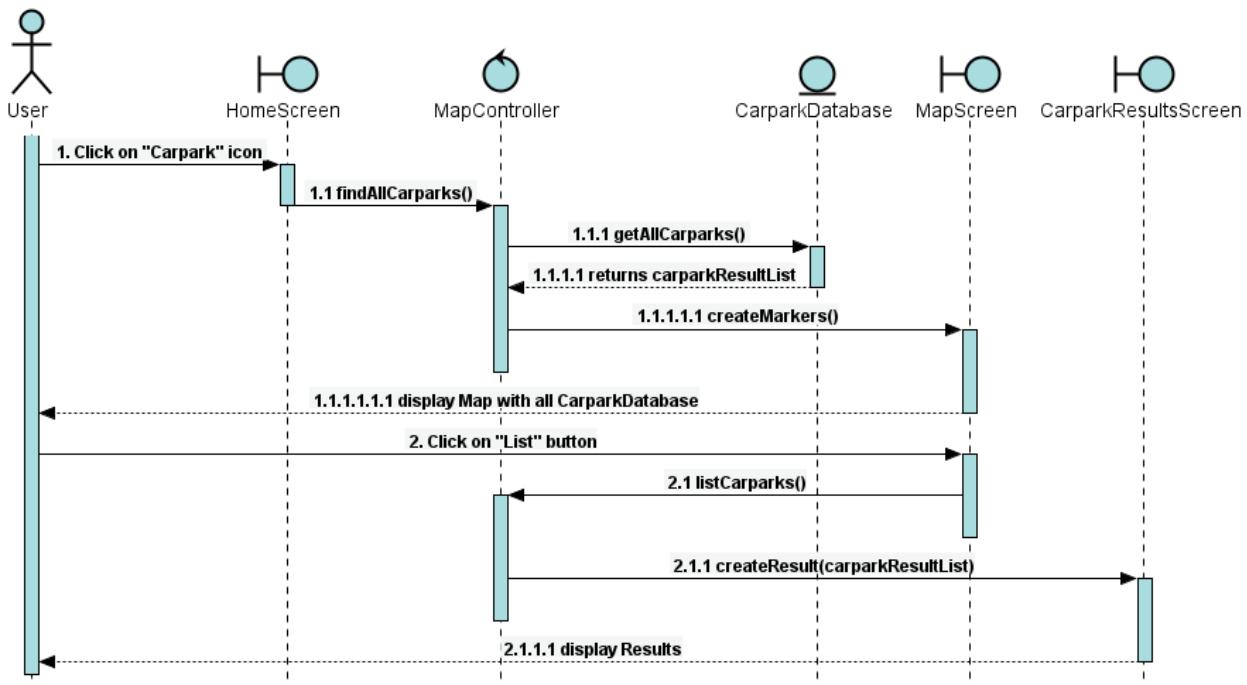


4.9 View Map of all Carparks - CZ009

Use Case ID:	CZ009		
Use Case Name:	View Map of all Carparks		
Created By:	Ananya B	Last Updated By:	Ananya B
Date Created:	24 Jan 2022	Date Last Updated:	20 Mar 2022

Actor:	User
Description:	User selects the carparks icon to view all the carparks as markers on a map as well as in a pop up list.
Preconditions:	<ol style="list-style-type: none"> 1. The user must have clicked the carparks icon. 2. The user must have an active internet connection 3. The user must have location services turned on and location permissions granted
Postconditions:	<ol style="list-style-type: none"> 1. User is able to spatially view all of the carparks as markers on a map. 2. The user can zoom in and out of the map by pinching.

	<p>3. User is able to click on “Show All Carparks” to view a list of carparks.</p> <ul style="list-style-type: none"> a. The user can view basic information about the carpark like name, type of carpark and whether there is free parking. b. The user can click on “Google Maps” to open google maps. c. The user may extend to Use Case CZ007 after selecting a specific carpark.
Priority:	Medium - Viewing surrounding carparks would generally be used by those who are near their destination, and just want a carpark with an available slot.
Frequency of Use:	Medium. The user will access the carpark information when no information about hawker centres is required.
Flow of Events:	<ol style="list-style-type: none"> 1. The user can view all carparks in Singapore as markers on the map after selecting the carpark icon next to the home page icon. 2. The user can click on the “List” button to view all carparks in a list. 3. User can click on “Google Maps” to get redirected to google maps with directions. 4. User may extend to Use Case CZ007 after selecting a specific carpark from the list.
Alternative Flows:	<p>AF-S1: If location service is turned off or location service permission is not granted</p> <ol style="list-style-type: none"> 1. The application will prompt the user for location services to be turned on and request for location services permission.
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

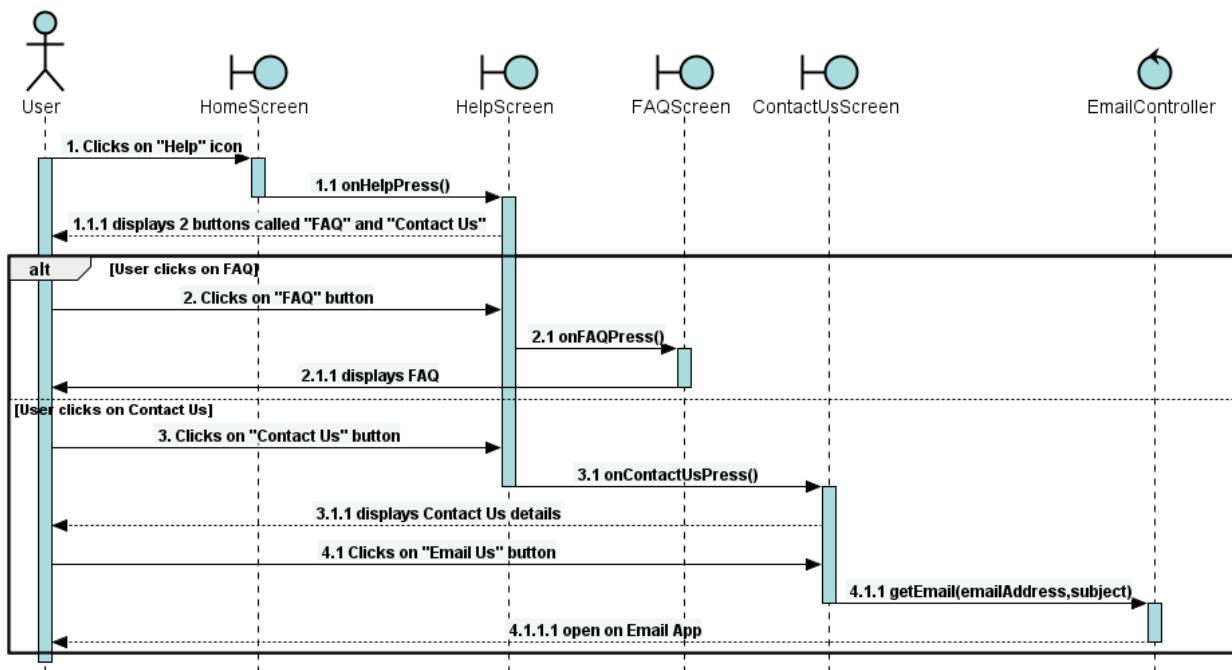


4.10 View Help Page - CZ010

Use Case ID:	CZ010		
Use Case Name:	View Help Page		
Created By:	Roshan	Last Updated By:	Roshan
Date Created:	22 Jan 2022	Date Last Updated:	25 Mar 2022

Actor:	User
Description:	User selects the 'Help' icon to access information on how to navigate the app and its various functions
Preconditions:	<ol style="list-style-type: none"> 1. The user's device needs to be currently connected to the internet 2. The user's device needs to enable location services under their privacy settings

	<p>3. The user needs to press the help icon next to the carparks icon to view the help page.</p>
Postconditions:	<ol style="list-style-type: none"> 1. The user can view two buttons, “FAQ” and “Contact Us” 2. The user can click on the “FAQ” button to view and access a list of Frequently Asked Questions on how to navigate the app and its various functions. 3. The user can click on the “Contact Us” button to contact us through email regarding any unresolved queries.
Priority:	Medium. ‘Request for Assistance’ feature serves as a guide to new users on how to use the various app functions and to voice out any issues, enabling an easier and smoother user experience.
Frequency of Use:	Low. Over time, the user will become more familiar with how to navigate the app, using this feature less often and only when they have issues.
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects the ‘Request for Assistance’ feature with the question mark icon in the ‘Home’ page 2. The user can either view and access a list of Frequently Asked Questions on how to navigate the app and its various functions or contact us through email regarding any unresolved queries
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL



5. Other Nonfunctional Requirements

5.1 Performance Requirements

- 5.1.1 Upon opening the application, the application's home page must be loaded within 3 seconds.
- 5.1.2 After entering the filter conditions, the list of matched hawker centres must load within 5 seconds.
- 5.1.3 The application must retrieve accurate carpark vacancies every 5 minutes.
- 5.1.4 The application must be able to return the nearest carparks and their respective routes within 3 seconds after the user enters their desired hawker centre they want to visit.
- 5.1.5 The application must remove all filters within 1 user click on the button.
- 5.1.6 The user must be able to zoom in and out of the map by clicking on the screen
- 5.1.7 The application must show the map with the markers accurately at the correct locations
- 5.1.8 The application must be able to store and display the 10 most recently viewed stalls correctly.
- 5.1.9 The application must be able to open the email.

5.2 Safety Requirements

- 5.2.1 The application must not require any personal information of users and should not save the user's history of actual hawker centre visits.
- 5.2.2 The application must not store the users' locations or track their locations for any other purpose other than the application's stated functionalities.

5.3 Security Requirements

- 5.3.1 The application does not require any user's information
- 5.3.2 The application uses AWS Elastic Beanstalk which provides AWS Identity and Access Management (IAM) managed policies for instance profiles, service roles, and IAM users.
- 5.3.3 The application uses AWS Elastic Beanstalk's Amazon CloudWatch Monitoring service which preserves reliability and security.

5.4 Software Quality Attributes

5.4.1 Usability

- 5.4.1.1 The user interface must be GUI based.
- 5.4.1.2 The user must be able to enter the desired filters in 1 attempt.
- 5.4.1.3 The user must be able to easily determine what a feature is and what it can do in 1 attempt.
- 5.4.1.4 The user is required to input the data from the cell phone keypad.
- 5.4.1.5 The system must provide necessary feedback to the user within 3 seconds when invalid inputs are detected.
- 5.4.1.6 To display an appropriate error message within 3 seconds when a certain process fails.
- 5.4.1.7 If network connection is unavailable, the application must prompt the user to try again with a network connection within 2 seconds.
- 5.4.1.8 The user must be able to zoom in and out of the map by pinching the screen.

5.4.2 Reliability

- 5.4.2.1 The application must be fully operational.
- 5.4.2.2 The application must not crash when the user has the application opened.
- 5.4.2.3 The application must have regular bug fix checks and the bug must be fixed immediately if the user has contacted us to complain about it.
- 5.4.2.4 The application must not crash while the user is sending an email

5.4.3 Scalability

- 5.4.3.1 The application must be extendable with new functionality without any disruption of existing activities.
- 5.4.3.2 The databases used by the application must be scalable to allow for additional rows of information
- 5.4.3.3 The filter function must be extendable to easily allow additional filter categories and options to be added to the application
- 5.4.3.4 The carpark and hawker stall information screen must be extendable to easily allow additional information to be displayed on the screen.

5.4.4 Supportability

- 5.4.4.1 The application must be compatible with phones running a specified version and above.

5.4.5 Extensibility

- 5.4.5.1 The application must be extensible as new features can be added to the application without disrupting existing activities.

5.5 Business Rules

- 5.5.1 All individuals can perform all functions.

6. Other Requirements

6.1 Reuse Objectives

We utilise service-oriented systems by linking shared services within our application that are externally provided. When users want to view the directions to their destination via their chosen route, they are brought to Google Maps for the routing service. This is done through different ways such as upon the clicking of a “Open in Google Maps” Button or Alert.

6.2 MVC framework

The application interface should follow the Model-View-Controller (MVC) model for rendering and modelling data objects. The interface must be able to connect to a database to retrieve the hawker stall information and retrieve live carpark data from Data.gov. Source and destination formats for the data includes JavaScript Object Notation (JSON) and Comma Separated Value (CSV).

7. Appendix

7.1: Appendix A: Data Dictionary

7.1.1 Hawker Centres

Attribute Name	Required	Format	Max Field Size	Location
Id	Yes	int	10	/swe/hawkerstalls.csv
Thumbnail	Yes	text	500	/swe/hawkerstalls.csv
Name	Yes	text	200	/swe/hawkerstalls.csv
Rating	Yes	int	10	/swe/hawkerstalls.csv
BusinessName	Yes	text	200	/swe/hawkerstalls.csv
UnitNo	Yes	text	20	/swe/hawkerstalls.csv
PhoneNumber	Yes	text	50	/swe/hawkerstalls.csv
OperationHours	Yes	text	200	/swe/hawkerstalls.csv
CommunityRating	Yes	int	10	/swe/hawkerstalls.csv
HighlightItems	Yes	text	50,000	/swe/hawkerstalls.csv
FoodCategories	Yes	text	10,000	/swe/hawkerstalls.csv

7.1.2 Hawker Centres Closure

Attribute Name	Required	Format	Max Field Size	Location
serial_no	Yes	int	10	/swe/hawker_closure.csv
name	Yes	text	200	/swe/hawker_closure.csv
_id	Yes	int	10	/swe/hawker_closure.csv
q1_cleaningstartdate	Yes	date	20	/swe/hawker_closure.csv
q1_cleaningenddate	Yes	date	20	/swe/hawker_closure.csv
remarks_q1	Yes	text	100	/swe/hawker_closure.csv
q2_cleaningstartdate	Yes	date	20	/swe/hawker_closure.csv
q2_cleaningenddate	Yes	date	20	/swe/hawker_closure.csv

remarks_q2	Yes	text	100	/swe/hawker_closure.csv
q3_cleaningstartdate	Yes	date	20	/swe/hawker_closure.csv
q3_cleaningenddate	Yes	date	20	/swe/hawker_closure.csv
remarks_q3	Yes	text	100	/swe/hawker_closure.csv
q4_cleaningstartdate	Yes	date	20	/swe/hawker_closure.csv
q4_cleaningenddate	Yes	date	20	/swe/hawker_closure.csv
remarks_q4	Yes	text	100	/swe/hawker_closure.csv
other_works_startdate	Yes	date	20	/swe/hawker_closure.csv
other_works_enddate	Yes	date	20	/swe/hawker_closure.csv
remarks_other_works	Yes	text	100	/swe/hawker_closure.csv
latitude_hc	Yes	float	20	/swe/hawker_closure.csv
longitude_hc	Yes	float	20	/swe/hawker_closure.csv
photourl	Yes	text	500	/swe/hawker_closure.csv
address_myenv	Yes	text	500	/swe/hawker_closure.csv
no_of_market_stalls	Yes	int	10	/swe/hawker_closure.csv
no_of_food_stalls	Yes	int	10	/swe/hawker_closure.csv
description_myenv	Yes	text	20,000	/swe/hawker_closure.csv
status	Yes	text	50	/swe/hawker_closure.csv
google_3d_view	Yes		200	/swe/hawker_closure.csv
google_for_stall	Yes	text	100	/swe/hawker_closure.csv

7.1.3 Carpark Closure

Attribute Name	Required	Format	Max Field Size	Location
id	Yes	int	10	/swe/parkingAvailability.csv
carpark_number	Yes	text	10	/swe/parkingAvailability.csv
update_datetime	Yes	date	50	/swe/parkingAvailability.csv
total_lots	Yes	int	10	/swe/parkingAvailability.csv
lot_type	Yes	char	1	/swe/parkingAvailability.csv

lots_available	Yes	int	10	/swe/parkingAvailability.csv
----------------	-----	-----	----	------------------------------

7.1.4 Carpark Code

Attribute Name	Required	Format	Max Field Size	Location
Id	Yes	int	10	/swe/code_carpark.csv
short_term_parking	Yes	text	50	/swe/code_carpark.csv
car_park_type	Yes	text	50	/swe/code_carpark.csv
y_coord	Yes	float	20	/swe/code_carpark.csv
x_coord	Yes	float	20	/swe/code_carpark.csv
free_parking	Yes	text	50	/swe/code_carpark.csv
gantry_height	Yes	float	10	/swe/code_carpark.csv
car_park_basement	Yes	char	1	/swe/code_carpark.csv
night_parking	Yes	text	10	/swe/code_carpark.csv
address	Yes	text	100	/swe/code_carpark.csv
car_park_decks	Yes	int	10	/swe/code_carpark.csv
_id	Yes	int	10	/swe/code_carpark.csv
car_park_no	Yes	text	20	/swe/code_carpark.csv
type_of_parking_system	Yes	text	50	/swe/code_carpark.csv

7.1.5 Google Maps Routing API

Attribute Name	Required	Format	Max Field Size	Location
origin	Yes	String	1	/swe/code_goolglemaps_routing.csv
destination	Yes	String	1	/swe/code_goolglemaps_routing.csv
mode	Yes	String	1	/swe/code_goolglemaps_routing.csv
departure_time	Yes	String	1	/swe/code_goolglemaps_routing.csv
arrival_time	Yes	String	1	/swe/code_goolglemaps_routing.csv

waypoints	Yes	String	1	/swe/code_googlemaps_routing.csv
optimise_waypoints	Yes	Boolean	1	/swe/code_googlemaps_routing.csv
alternatives	No	Boolean	1	/swe/code_googlemaps_routing.csv
avoid	No	Char	1	/swe/code_googlemaps_routing.csv
units	Yes	String	1	/swe/code_googlemaps_routing.csv
traffic_model	No	String	1	/swe/code_googlemaps_routing.csv
transit_mode	Yes	String	3	/swe/code_googlemaps_routing.csv
transit_routing_preference	Yes	String	1	/swe/code_googlemaps_routing.csv
language	Yes	String	1	/swe/code_googlemaps_routing.csv
region	Yes	String	1	/swe/code_googlemaps_routing.csv
key	Yes	String	1	/swe/code_googlemaps_routing.csv
simplify	No	Boolean	1	/swe/code_googlemaps_routing.csv
curl_proxy	Yes	curl_proxy	1	/swe/code_googlemaps_routing.csv

7.2 Appendix B: Testing

7.2.1 BlackBox Testing - Search Controller Class

7.2.1.1 Equivalence Class and Boundary Values

The class takes in 1 input from the user:

1. The input is a string and must either match
 - a. The food category of a hawker stall OR
 - b. The name of a hawker stall OR
 - c. The name of a hawker centre.

Parameter	Equivalence Class (EC)	Boundary Value
Search string entered	Valid EC: <ul style="list-style-type: none"> - Substring of a food category - Substring of a hawker stall name - Substring of a hawker center name Invalid EC: <ul style="list-style-type: none"> - Any phrase that does not fall under the 3 listed subsets 	Not applicable as there is no numerical range of values

In the case of Hawkerpedia, no effective Boundary Value Black Box Testing can be conducted as apart from user clicks, only the search has a wide range of user input. Hawkerpedia's Filter function is highly controlled due to the use of checkboxes.

7.2.1.2 Expected Results

Valid Input

Test Case ID	String Input	Substring of some Food Category	Substring of some Hawker Stalls	Substring of some Hawker Centre	Expected Output	Actual Output	Result
BB1.1 .1	Bak Chor Mee	True (Bak Chor Mee)	False	False	List of Hawker Stalls that sell Bak Chor Mee	List of Hawker Stalls that sell Bak Chor Mee	Passed
BB1.1 .2	Chor	True (Bak Chor Mee)	False	False	List of Hawker Stalls that sell Bak Chor Mee	List of Hawker Stalls that sell Bak Chor Mee	Passed

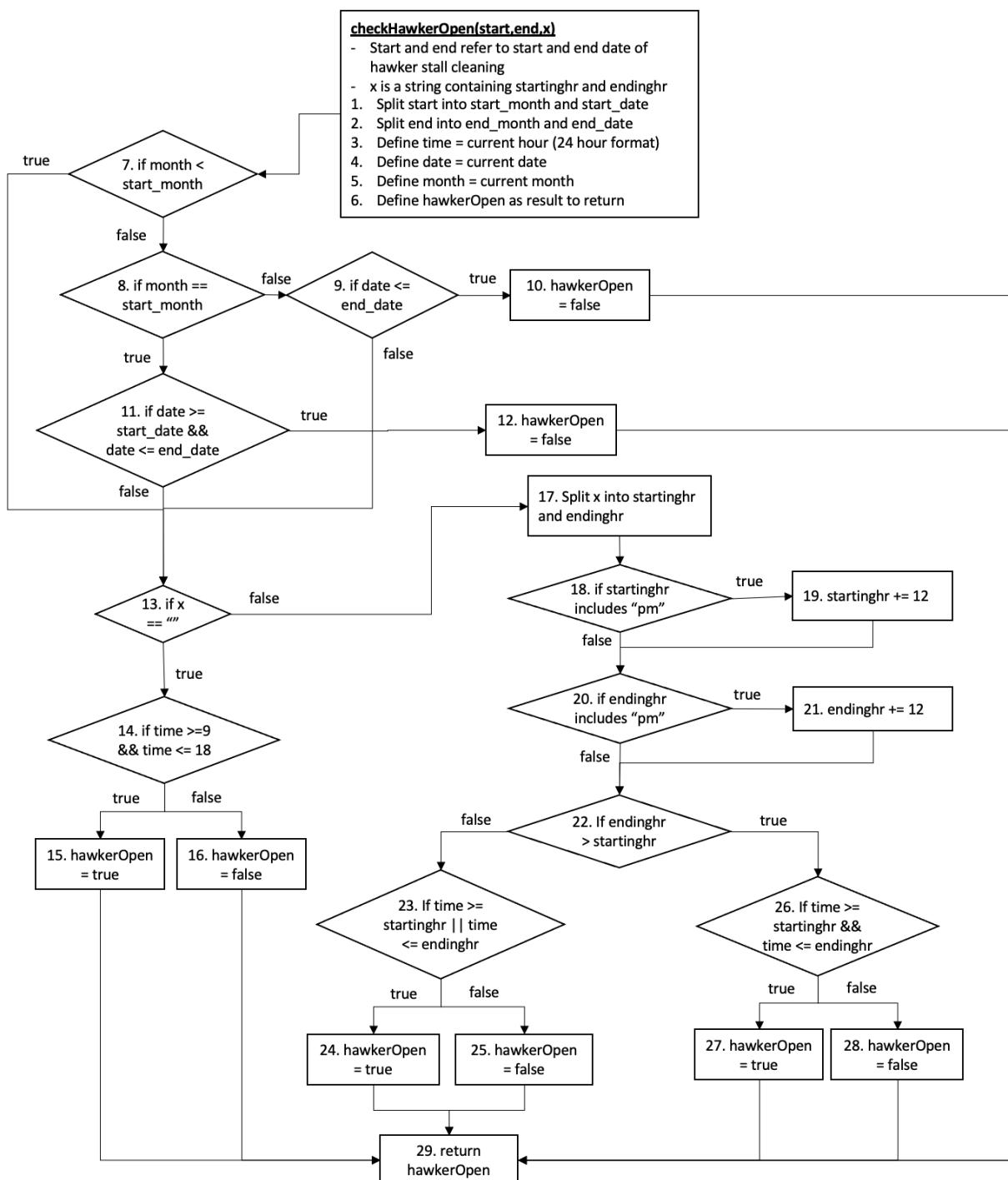
BB1.1 .3	Pondok	False	True (Pondok Satay)	False	Hawker Stall named Pondok Satay	Hawker Stall named Pondok Satay	Passed
BB1.1 .4	Pond	False	True (Pondok Satay)	False	Hawker Stall named Pondok Satay	Hawker Stall named Pondok Satay	Passed
BB1.1 .5	Aljunied Ave 2 Blk 117	False	False	True (Aljunied Ave 2 Blk 117)	List of Hawker Stalls that at Aljunied Ave 2 Blk 117 Hawker Centre	List of Hawker Stalls that at Aljunied Ave 2 Blk 117 Hawker Centre	Passed
BB1.1 .6	Blk 117	False	False	True (Aljunied Ave 2 Blk 117)	List of Hawker Stalls that at Aljunied Ave 2 Blk 117 Hawker Centre	List of Hawker Stalls that at Aljunied Ave 2 Blk 117 Hawker Centre	Passed
BB1.1 .7	Chicken Rice	True (Chicken Rice)	True (Chicken Rice)	False	List of Hawker Stalls that sell Chicken Rice or have Chicken Rice in their name	List of Hawker Stalls that sell Chicken Rice or have Chicken Rice in their name	Passed
BB1.1 .8	Chick	True (Chicken Rice)	True (Chicken Rice)	False	List of Hawker Stalls that sell Chicken Rice or have Chicken Rice in their name	List of Hawker Stalls that sell Chicken Rice or have Chicken Rice in their name	Passed
BB1.1 .9	Adam	False	True (Stalls at Adam Road)	True (Adam Road Food Centre)	List of Hawker Stalls that are at Adam Road Food Centre or have Adam in their name	List of Hawker Stalls that are at Adam Road Food Centre or have Adam in their name	Passed

7.2.1.3 Invalid Input

Test Case ID	String Input	Substring of some Food Category	Substring of some Hawker Stalls	Substring of some Hawker Centre	Expected Output	Actual Output	Result
BB1.2 .1	Dragon	False	False	False	No Results Found	No Results Found	Passed

BB1.2 .2	Software	False	False	False	No Results Found	No Results Found	Passed
BB1.2 .3	123456	False	False	False	No Results Found	No Results Found	Passed
BB1.2 .4	./	False	False	False	No Results Found	No Results Found	Passed
BB1.2 .5	abcd	False	False	False	No Results Found	No Results Found	Passed

7.2.2 Whitebox Testing - checkHawkerOpen()



<u>White Box Testing - checkHawkerOpen() Method</u>		
Cyclomatic Complexity		
$CC = \text{edges} - \text{nodes} + 2 = 34 - 24 + 2 = 12$		
$CC = \text{decisionpoint} + 1 = 11 + 1 = 12$		
Basis Paths		
#Path	Basis Path	Description of Basis Path Choice
1	1-6, 7, 8, 9, 13, 17, 18, 20, 21, 22, 26, 27, 29	Baseline path
2	1-6, 7, 13, 17, 18, 20, 21, 22, 26, 27, 29	Change first decision point from baseline path at node 7, testing 7 -> 13 edge
3	1-6, 7, 13, 14, 15, 29	Change second decision point at node 13 from path #2, testing node 15
4	1-6, 7, 13, 14, 16, 29	Change third decision point at node 14 from path #3, testing node 16
5	1-6, 7, 8, 11, 13, 17, 18, 20, 21, 22, 26, 27, 29	Change second decision point at node 14 from baseline path, testing node 11
6	1-6, 7, 8, 11, 12, 29	Change third decision point at node 11 from path #5, testing node 12
7	1-6, 7, 8, 9, 10, 29	Change third decision point at node 9 from baseline path, testing node 10
8	1-6, 7, 8, 9, 13, 17, 18, 19, 20, 21, 22, 26, 27, 29	Change fifth decision point at node 18 from baseline path, testing node 19
9	1-6, 7, 8, 9, 13, 17, 18, 19, 20, 22, 26, 27, 29	Change sixth decision point at node 20 from path #8 → infeasible
10	1-6, 7, 8, 9, 13, 17, 18, 19, 20, 22, 23, 24, 29	Change sixth decision point at node 20 and 7th decision point at node #22 from path #8, testing node 24 → double change of decision points to ensure feasibility
11	1-6, 7, 8, 9, 13, 17, 18, 19, 20, 22, 23, 25, 29	Change 8th decision point at node 23 from path #10, testing node 25

12	1-6, 7, 8, 9, 13, 17, 18, 20, 21, 22, 26, 28, 29	Change 8th decision point at node 26 from baseline path, testing node 28
Test cases		
#Path	Description of Test Case	Actual values of Test Case
1	Baseline path: Current month is after cleaning start month, current date not within cleaning duration, details of opening hours are available, starting hour is not past noon, ending hour is past noon, hawker does not operate past midnight, current time is within opening hours	Current Date = 09/04/2022 Current Time = 4pm Start = 02/03/2022 End = 03/03/2022 X (operation hour) = 10am - 5pm
2	Current month is before cleaning start month , details of opening hours are available, starting hour is not past noon, ending hour is past noon, hawker does not operate past midnight, current time is within opening hours.	Current Date = 09/04/2022 Current Time = 4pm Start = 02/05/2022 End = 03/05/2022 X (operation hour) = 10am - 5pm
3	Current month is before cleaning start month, details of opening hours are not available , current time is within 9am and 6pm.	Current Date = 09/04/2022 Current Time = 4pm Start = 02/05/2022 End = 03/05/2022 X (operation hour) = “”
4	Current month is before cleaning start month, details of opening hours are not available, current time is not within 9am and 6pm	Current Date = 09/04/2022 Current Time = 8pm Start = 02/05/2022 End = 03/05/2022 X (operation hour) = “”
5	Current month is on cleaning start month , current date not within cleaning duration, details of opening hours are available, starting hour is not past noon, ending hour is past noon, hawker does not operate past midnight, current time is within opening hours	Current Date = 09/04/2022 Current Time = 4pm Start = 02/04/2022 End = 03/04/2022 X (operation hour) = 10am - 5pm
6	Current month is on cleaning start month, current date is within cleaning	Current Date = 09/04/2022 Current Time = 4pm

	duration.	Start = 08/04/2022 End = 10/04/2022 X (operation hour) = 10am - 5pm
7	Current month is after cleaning start month , current date is within cleaning duration.	Current Date = 01/04/2022 Current Time = 4pm Start = 31/03/2022 End = 2/04/2022 X (operation hour) = 10am - 5pm
8	Current month is after cleaning start month, current date not within cleaning duration, details of opening hours are available, starting hour is past noon , ending hour is past noon, hawker does not operate past midnight, current time is within opening hours	Current Date = 09/04/2022 Current Time = 4pm Start = 02/03/2022 End = 03/03/2022 X (operation hour) = 1pm - 9pm
9	Current month is after cleaning start month, current date not within cleaning duration, details of opening hours are available, starting hour is past noon, ending hour is not past noon , hawker does not operate past midnight, current time is within opening hours (infeasible)	Current Date = 09/04/2022 Current Time = 4pm Start = 02/03/2022 End = 03/03/2022 X (operation hour) = 1pm - 1am (Hawker doesn't operate past midnight) (Infeasible)
10	Current month is after cleaning start month, current date not within cleaning duration, details of opening hours are available, starting hour is past noon, ending hour is not past noon , hawker operates past midnight , current time is within opening hours (change both decision points at same time)	Current Date = 09/04/2022 Current Time = 4pm Start = 02/03/2022 End = 03/03/2022 X (operation hour) = 1pm - 1am (Hawker operates past midnight) (change 2 decision points)
11	Current month is after cleaning start month, current date not within cleaning duration, details of opening hours are available, starting hour is past noon, ending hour is not past noon, hawker operates past midnight, current time is not within opening hours	Current Date = 09/04/2022 Current Time = 11am Start = 02/03/2022 End = 03/03/2022 X (operation hour) = 1pm - 1am (Hawker operates past midnight)
12	Current month is after cleaning start	Current Date = 09/04/2022

	month, current date not within cleaning duration, details of opening hours are available, starting hour is not past noon, ending hour is past noon, hawker does not operate past midnight, current time is not within opening hours	Current Time = 8pm Start = 02/03/2022 End = 03/03/2022 X (operation hour) = 10am - 5pm
--	--	--

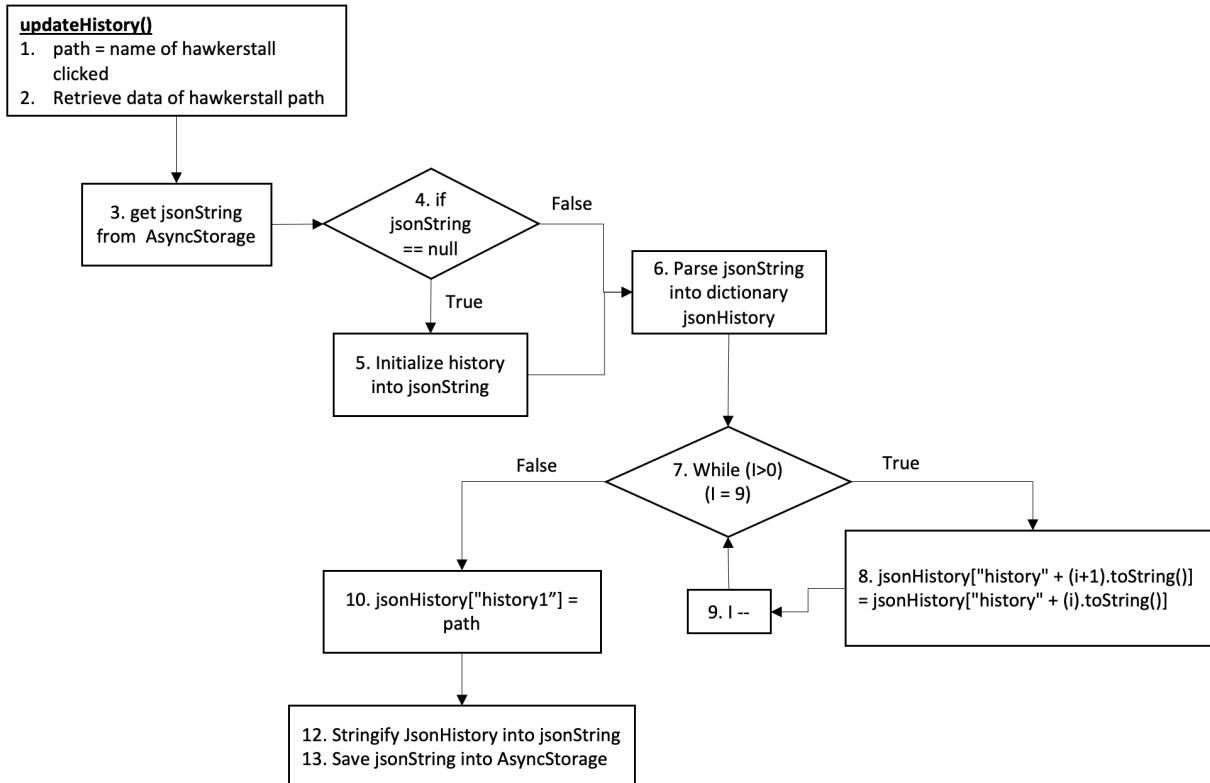
Real Execution Paths

#Path	Real Executed Path	Description of Result
1	1-6, 7, 8, 9, 13, 17, 18, 20, 21, 22, 26, 27, 29	Current Date = 09/04/2022 Current Time = 4pm Start = 02/03/2022 End = 03/03/2022 X (operation hour) = 10am - 5pm ⇒ Hawker Open
2	1-6, 7, 13, 17, 18, 20, 21, 22, 26, 27, 29	Current Date = 09/04/2022 Current Time = 4pm Start = 02/05/2022 End = 03/05/2022 X (operation hour) = 10am - 5pm ⇒ Hawker Open
3	1-6, 7, 13, 14, 15, 29	Current Date = 09/04/2022 Current Time = 4pm Start = 02/05/2022 End = 03/05/2022 X (operation hour) = "" ⇒ Hawker Open
4	1-6, 7, 13, 14, 16, 29	Current Date = 09/04/2022 Current Time = 8pm Start = 02/05/2022 End = 03/05/2022 X (operation hour) = "" ⇒ Hawker Closed
5	1-6, 7, 8, 11, 13, 17, 18, 20, 21, 22, 26, 27, 29	Current Date = 09/04/2022 Current Time = 4pm Start = 02/04/2022 End = 03/04/2022 X (operation hour) = 10am - 5pm ⇒ Hawker Open

6	1-6, 7, 8, 11, 12, 29	Current Date = 09/04/2022 Current Time = 4pm Start = 08/04/2022 End = 10/04/2022 X (operation hour) = 10am - 5pm ⇒ Hawker Closed
7	1-6, 7, 8, 9, 10, 29	Current Date = 01/04/2022 Current Time = 4pm Start = 31/03/2022 End = 2/04/2022 X (operation hour) = 10am - 5pm ⇒ Hawker Closed
8	1-6, 7, 8, 9, 13, 17, 18, 19, 20, 21, 22, 26, 27, 29	Current Date = 09/04/2022 Current Time = 4pm Start = 02/03/2022 End = 03/03/2022 X (operation hour) = 1pm - 9pm ⇒ Hawker Open
9	Infeasible Basis Path	
10	1-6, 7, 8, 9, 13, 17, 18, 19, 20, 22, 23, 24, 29	Current Date = 09/04/2022 Current Time = 4pm Start = 02/03/2022 End = 03/03/2022 X (operation hour) = 1pm - 1am (Hawker operates past midnight) ⇒ Hawker Open
11	1-6, 7, 8, 9, 13, 17, 18, 19, 20, 22, 23, 25, 29	Current Date = 09/04/2022 Current Time = 11am Start = 02/03/2022 End = 03/03/2022 X (operation hour) = 1pm - 1am (Hawker operates past midnight) ⇒ Hawker Closed
12	1-6, 7, 8, 9, 13, 17, 18, 20, 21, 22, 26, 28, 29	Current Date = 09/04/2022 Current Time = 8pm Start = 02/03/2022 End = 03/03/2022 X (operation hour) = 10am - 5pm

		⇒ Hawker Closed
--	--	-----------------

7.2.3 Whitebox Testing - updateHistory()



<u>White Box Testing - updateHistory() Method</u>		
Cyclomatic Complexity		
$CC = edges - nodes + 2 = 10 - 9 + 2 = 3$ $CC = decisionpoint + 1 = 2 + 1 = 3$		
Basis Paths		
#Path	Basis Path	Description of Basis Path Choice
1	1-2, 3, 4, 6, 7, 10, 12-13	Baseline path
2	1-2, 3, 4, 5, 6, 7, 10, 12-13	Change first decision point from baseline path, test

		node 5
3	1-2, 3, 4, 6, 7, 8, 9, 7, 10, 12-13	Selecting while loop once

Test cases

#Path	Description of Test Case	Actual values of Test Case
1	Baseline path: Update History when history has previously been instantiated	path = Jia Le Spinach Soup history = {"history1": "Mr Avocado", "history2": "Fu He Delight", "history3": "Leon Kee Claypot Pork Rib Soup", "history4": "Fu He Turtle Soup", "history5": "King Avocado", "history6": "Chwee Kueh Bukit Timah Branch", "history7": "Zhou Pin", "history8": "3 Hainanese Chicken Rice", "history9": "Hock Kee Fried Kway Teow", "history10": "Zhen Yuan"}
2	Update history when history has not been instantiated before	path = Jia Le Spinach Soup history = ""
3	Update history when history has previously been instantiated, with shifting of older history	path = Jia Le Spinach Soup history = {"history1": "Mr Avocado", "history2": "Fu He Delight", "history3": "Leon Kee Claypot Pork Rib Soup", "history4": "Fu He Turtle Soup", "history5": "King Avocado", "history6": "Chwee Kueh Bukit Timah Branch", "history7": "Zhou Pin", "history8": "3 Hainanese Chicken Rice", "history9": "Hock Kee Fried Kway Teow", "history10": "Zhen Yuan"}

Real Execution Paths

#Path	Real Executed Path	Description of Result
1	1-2, 3, 4, 6, 7, 8, 9, 7, 8, 9, 7, 8, 9, 7, 8, 9, 7, 8, 9, 7, 8, 9, 7, 10, 12-13	history = {"history1": "Jia Le Spinach Soup", "history2": "Mr Avocado", "history3": "Fu He Delight", "history4": "Leon Kee Claypot Pork Rib Soup", "history5": "Fu He Turtle Soup", "history6": "Chwee Kueh Bukit Timah Branch", "history7": "Zhou Pin", "history8": "3 Hainanese Chicken Rice", "history9": "Hock Kee Fried Kway Teow", "history10": "Zhen Yuan"}

		"history6": "King Avocado", "history7": "Chwee Kueh Bukit Timah Branch", "history8": "Zhou Pin", "history9": "3 Hainanese Chicken Rice", "history10": "Hock Kee Fried Kway Teow"}
2	1-2, 3, 4, 5, 6, 7, 10, 12-13	history = {"history1": "Jia Le Spinach Soup"}
3	1-2, 3, 4, 6, 7, 8, 9, 7, 8, 9, 7, 8, 9, 7, 8, 9, 7, 8, 9, 7, 8, 9, 7, 10, 12-13	history = {"history1": "Jia Le Spinach Soup", "history2": "Mr Avocado", "history3": "Fu He Delight", "history4": "Leon Kee Claypot Pork Rib Soup", "history5": "Fu He Turtle Soup", "history6": "King Avocado", "history7": "Chwee Kueh Bukit Timah Branch", "history8": "Zhou Pin", "history9": "3 Hainanese Chicken Rice", "history10": "Hock Kee Fried Kway Teow"}

7.2.4 Other Testing - Scenario Based Testing

Test Case ID	Scenario	Expected Result	Actual Result	Result
TC01	User clicks on the Home Tab at the bottom bar of the application	The system displays the Home Page.	The Home Page is displayed	Passed
TC02	User clicks on a Hawker Stall Icon on the application's Home Page	The system displays the corresponding information page of the hawker stall	The correct information page of the hawker stall is displayed	Passed
TC03	User clicks on the back button after TC02	The system displays the Home Page, and updates the recent hawker list with the recently clicked hawker stall	The Home Page is displayed and the Recent list is updated accordingly	Passed
TC04	User clicks on the back button on any Screen	The system displays the previous screen that the User was on	The previous screen is displayed accordingly	Passed
TC05	User clicks on the search bar at the top of the application's Home Page	The system displays the Search Screen with suggested hawker stalls and their summarised information	The Search Screen is displayed with suggested hawker stalls	Passed
TC06	User clicks on a Hawker Stall Icon on the application's Search Screen	The system displays the corresponding information page of the hawker stall	The correct information page of the hawker stall is displayed	Passed
TC07	User starts to type a search keyword in the application's Search Screen	The system displays hawker stalls that match the substring typed	The correct hawker stalls buttons are displayed	Passed
TC08	User clicks the "x" button on the search	The system removes any string input in the	The search bar in the Search Screen is	Passed

	bar of the application's Search Screen	search bar	cleared accordingly	
TC09	User types a search keyword and clicks enter in the application's Search Screen	The system displays a list of hawker stalls that has a stall name, food category whose hawker centre matches the search keyword in the Results Screen. Summarised information and open status is also displayed	The correct hawker stalls with summarised information are displayed in the Results Screen,	Passed
TC10	User clicks Bus or the Walk icon in a hawker Stall's Information Screen	The system will open Google Maps with directions from the User's current location to the hawker stall.	Google Maps with the corresponding direction details are displayed	Passed
TC11	User clicks the Car icon in a hawker stall's Information Screen	The system will display Nearby Carpark Map Screen with nearby carparks within 1km of the chosen hawker stall	Nearby Carpark Map is displayed with pins marking nearby carparks	Passed
TC12	User double taps on the map of the Nearby Carpark Map	The system will display a zoom in view of the map	The magnified Nearby Carpark Map is displayed	Passed
TC13	User double clicks on the carpark pin on the Nearby Carpark Map	The system will display an alert to view route in google maps	The alert in Nearby Carpark Map is displayed	Passed
TC14	User clicks "Yes" when alert in Nearby Carpark Map is displayed	The system will open Google Maps with directions from the User's current location to the selected carpark	Google Maps with the corresponding direction details to the carpark are displayed	Passed
TC15	User clicks the Menu button in Nearby Carpark Map	The system will display a list of all nearby carparks that was displayed on the Nearby Carpark	The system displays a list of all nearby carparks accordingly	Passed

		Map, with summarised information such as lots available.		
TC16	User clicks on a carpark in the All Nearby Carparks menu of the Nearby carpark map	The system will display the corresponding additional information of the carpark, including live data on lots available.	The corresponding additional information of the carpark is displayed, including live data on lots available.	Passed
TC17	User clicks on the Google Maps button in the All Nearby Carparks menu of the Nearby carpark map	The system will open Google Maps with directions from the User's current location to the selected carpark	Google Maps with the corresponding direction details to the carpark are displayed	Passed
TC18	User types an invalid search keyword in the Search bar of the Search Screen	The system displays a "No results found" warning	"No results found" warning is displayed	Passed
TC19	User clicks on the filter button at the top of the Search Screen Page	The system displays a bottom sheet of the various filter checklists of Rating, Cuisine, Distance and Neighbourhood	The filter raw bottom sheet is displayed with the 4 filter checklists	Passed
TC20	User clicks on the checklist of any filter option on the Filter Sheet of the Search Screen	The system toggles the filter checklist option on/off depending on its previous state	The filter checklist option is toggled	Passed
TC21	User clicks on another Filter Ratings or Distance option when one option was already selected	The system will deselect the previously selected filter option as the filter options for ratings and distance are mutually exclusive	The filter checklist option is deselected accordingly	Passed
TC22	User clicks on a "Clear Filters" button on the Filter Screen	The system will deselect all filter checklist options belonging to that	The filter checklist options are deselected accordingly if they	Passed

	for Ratings, Cuisine, Distance or Neighbourhood	corresponding type of filter	were selected before.	
TC23	User clicks on the "Clear All" button on the Filter Sheet of the Search Screen	The system will deselect all filter checklist options.	All previously selected filter checklist options are deselected	Passed
TC24	User clicks on the "Done" button on the Filter Sheet of the Search Screen	The system will apply the selected filters to all hawker stalls and display the matching hawker stalls on the Results Page	The Results Page with the correspondingly filtered hawker stalls is displayed	Passed
TC25	User clicks on a Hawker Stall Icon on the application's Result Screen	The system displays the corresponding information page of the hawker stall	The correct information page of the hawker stall is displayed	Passed
TC26	User clicks on the Maps Tab at the bottom of the application	The system displays the Hawker Maps Page.	The Hawker Maps Page is displayed	Passed
TC27	User double taps on the map on the Hawker Map Screen	The system will display a zoom in view of the map on the Hawker Map Screen	The magnified Hawker Map Screen is displayed	Passed
TC28	User double clicks on the Hawker pin on the Hawker Map Screen	The system will display an alert to view route in google maps	The alert in Hawker Map Screen is displayed	Passed
TC29	User clicks "Yes" when the alert in Hawker Map Screen is displayed	The system will open Google Maps with directions from the User's current location to the selected Hawker Centre	Google Maps with the corresponding direction details to the Hawker Centre are displayed	Passed
TC30	User clicks the Menu button in Hawker Map Screen	The system will display a list of all nearby Hawker Centres that was	The system displays a list of all nearby Hawker Centres	Passed

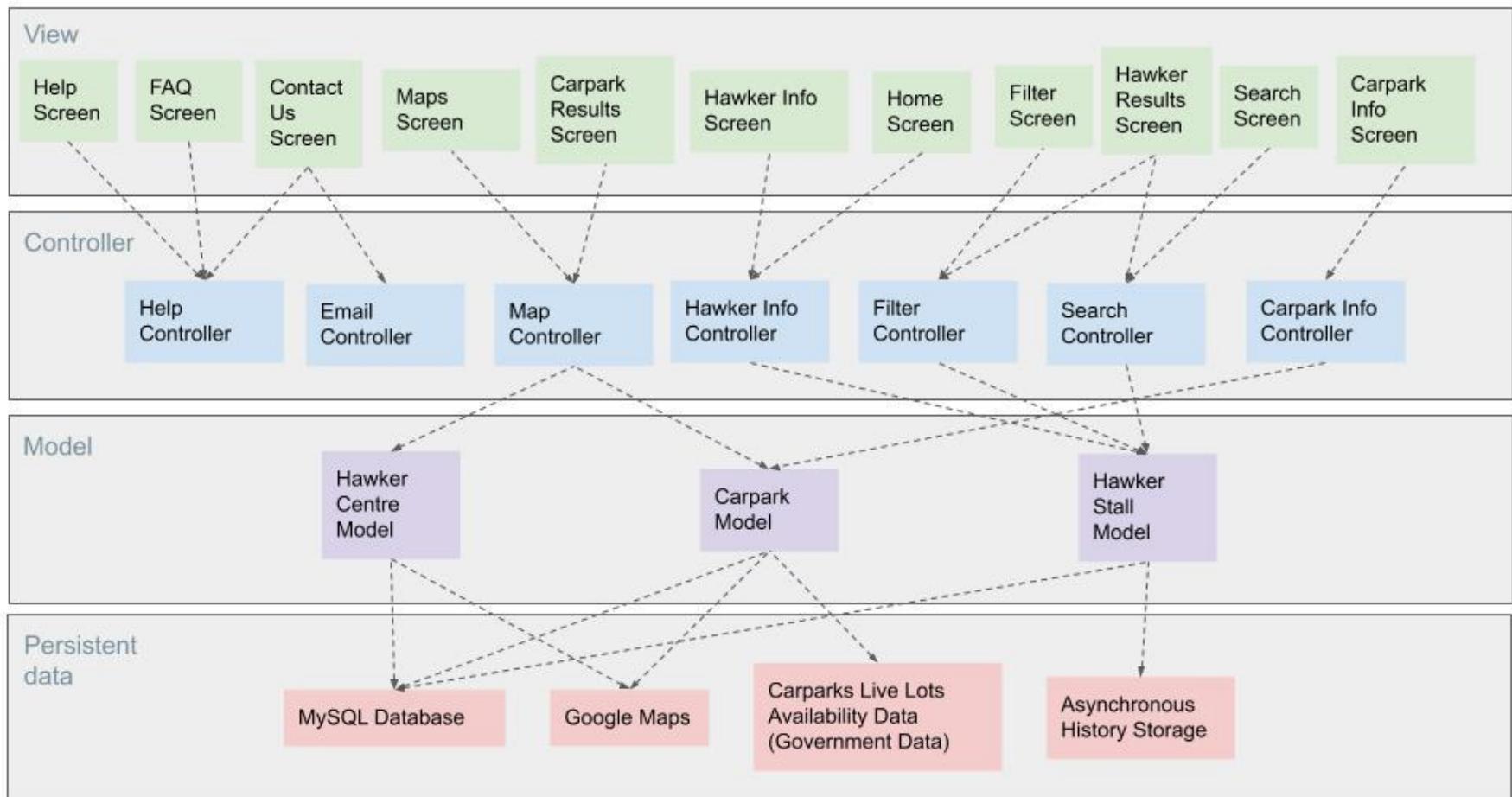
		displayed on the Hawker Map Screen, with summarised information including Hawker Centre Closure dates and Hawker Centre Open status.	accordingly	
TC31	User clicks on a Hawker Centre in the All Hawker Centres menu of the Hawker Map Screen	The system will display all Hawker Stalls of the selected Hawker Centre in a Results Screen	The result screen of Hawker Stalls is displayed	Passed
TC32	User clicks on the Google Maps button in the All Hawker Centres menu of the Hawker Map Screen	The system will open Google Maps with directions from the User's current location to the selected Hawker Centre	Google Maps with the corresponding direction details to the Hawker Centre are displayed	Passed
TC33	User clicks on the Carparks Tab at the bottom of the application	The system displays the Carpark Maps Page.	The Carpark Maps Page is displayed	Passed
TC34	User clicks and drags on the map on the Carpark Maps Screen	The system's map on the Carpark Maps Screen is moved around accordingly	The moved map in Carpark Map Screen is displayed	Passed
TC35	User double clicks on the Carpark pin on the Carpark Map Screen	The system will display an alert to view route in google maps	The alert in Hawker Map Screen is displayed	Passed
TC36	User clicks "Yes" when the alert in Carpark Map Screen is displayed	The system will open Google Maps with directions from the User's current location to the selected Hawker Centre	Google Maps with the corresponding direction details to the Hawker Centre are displayed	Passed
TC37	User clicks the Menu button in Carpark Map Screen	The system will display a list of all nearby Hawker Centres that was	The system displays a list of all nearby Hawker Centres	Passed

		displayed on the Hawker Map Screen, with summarised information including Hawker Centre Closure dates and Hawker Centre Open status.	accordingly	
TC38	User clicks on a carpark in the All Carparks menu of the Carpark Map Screen	The system will display the corresponding additional information of the carpark, including live data on lots available.	The corresponding additional information of the carpark is displayed, including live data on lots available.	Passed
TC39	User clicks on the Google Maps button in the All Nearby Carparks menu of the Carpark Map Screen	The system will open Google Maps with directions from the User's current location to the selected carpark	Google Maps with the corresponding direction details to the carpark are displayed	Passed
TC40	User clicks on the Help Tab at the bottom of the application	The system displays the Help Page.	The Help Page is displayed	Passed
TC41	User clicks on the FAQ button on the application's Help Page	The system displays the FAQ page with a list of all frequently asked questions.	The FAQ Page is displayed with frequently asked questions.	Passed
TC42	User clicks on the Contact Us button on the application's Help Page	The system displays the Contact Us page	The Contact Us Page is displayed	Passed
TC43	User clicks on the "Send Email" button on the application's Contact Us Page	The system displays a popup with the User's email, with the subject title of "Query to Hawkerpedia" automatically filled	An email is displayed with the relevant title filled.	Passed

7.3 Appendix C: Analysis Models

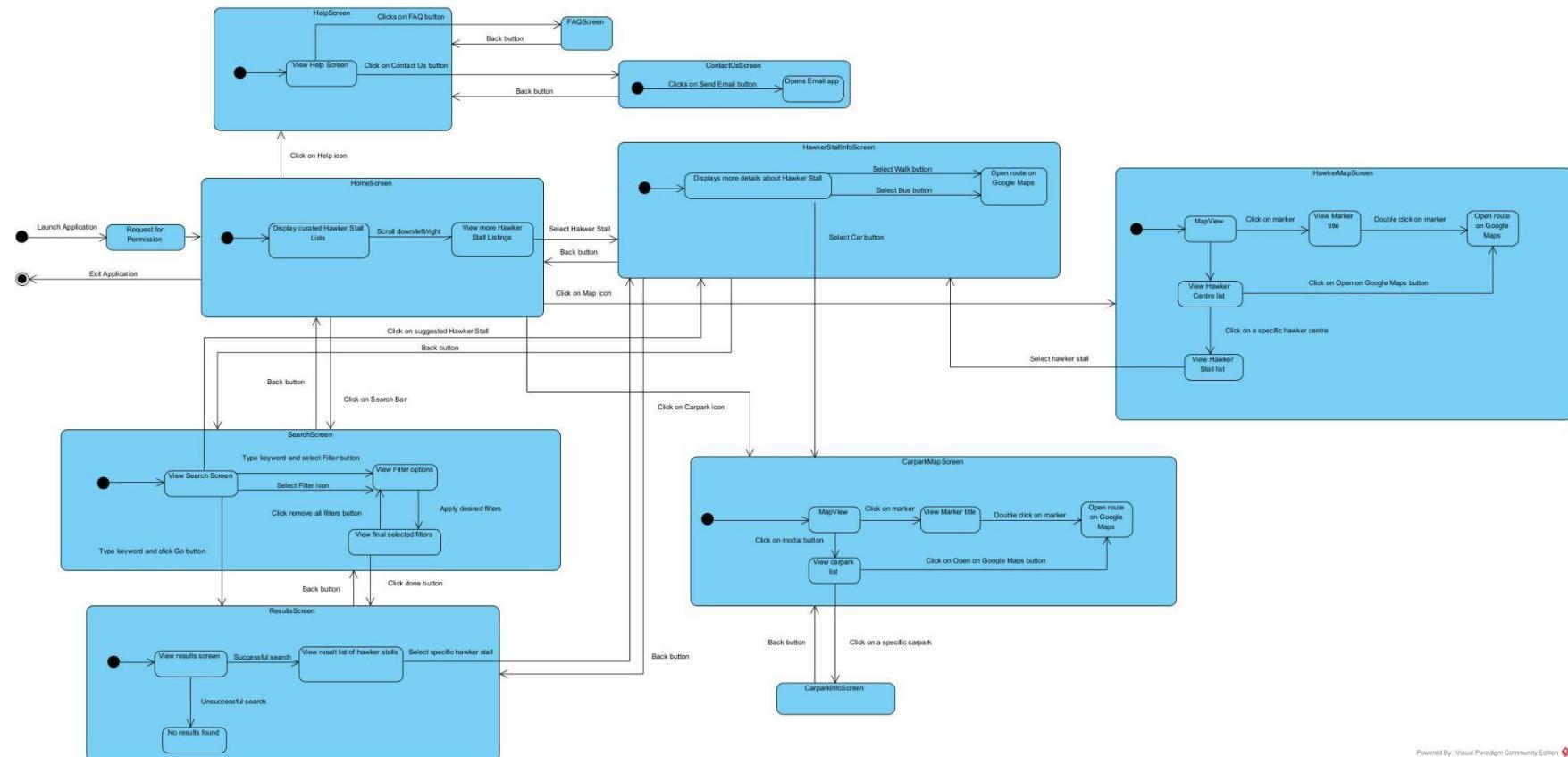
7.3.1 System Architecture Diagram

(See Architecture.jpg in Diagrams folder)



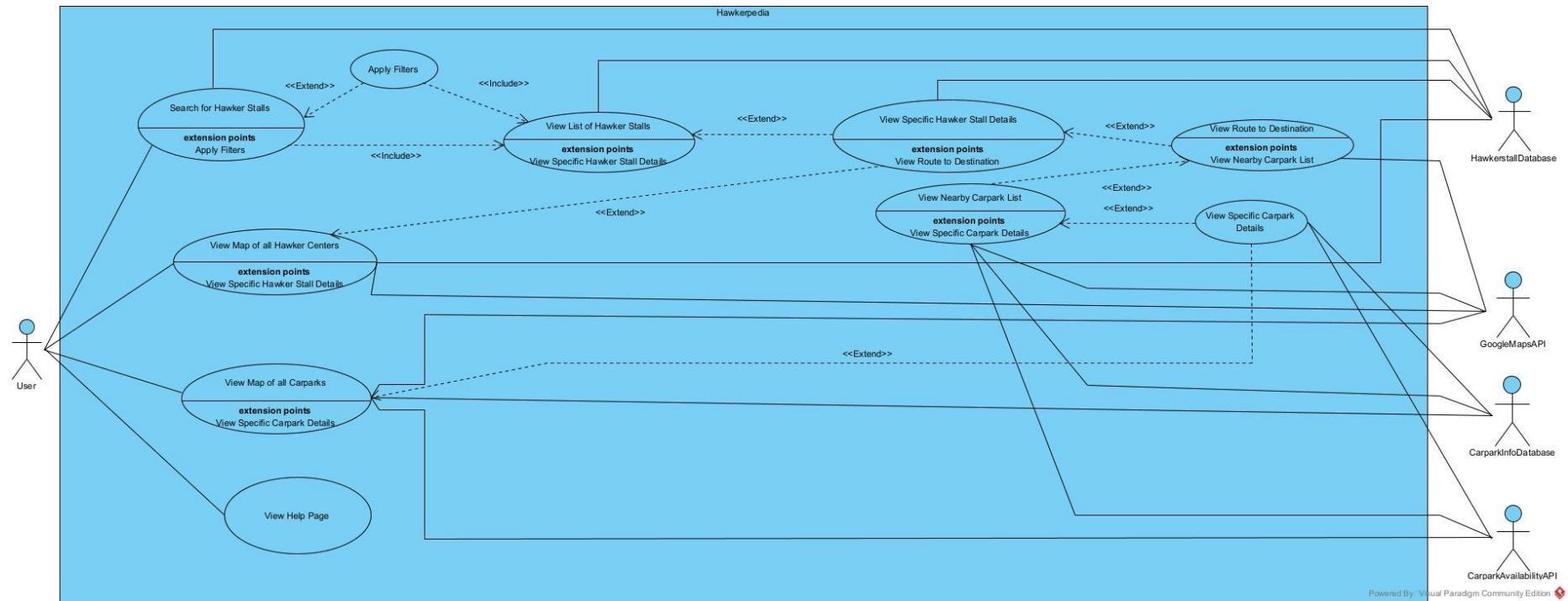
7.3.2 Dialog Map

(See DialogMap.jpg in Diagrams folder)



7.3.3 Use Case Diagram

(See Use Case Diagram.jpg in Diagrams folder)



7.3.4 Class Diagram

(See Class Diagram.jpg in Diagrams folder)

