Software Requirements Specification

For

MozzyShield

Version 1.0 approved

Prepared by Group 5

SOH QIAN YI

SEOW JING HNG ALOYSIUS

ZETA CHUA HUI SHI

SUI LULU

JOLENE TAN

ARAVIND S/O SIVAKUMARAN

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this project is to increase awareness towards the prevailing dengue situation in Singapore through an application called “MozzyShield”, by providing users with relevant information for dengue prevention.

1.2 Document Conventions

This document is based on the IEEE’s Software Requirements Specification template, and therefore follows the conventions set out by the IEEE.

1.3 Intended Audience and Reading Suggestions

This document is intended for users and developers who wish to understand the main functionalities of our application. The following sections will describe the motivation for this project, the functional and non-functional requirements. This document is read top-down.

1.4 Product Scope

MozzyShield is a mobile application to raise awareness for the prevailing dengue cases in Singapore.

The application provides live dengue cluster monitoring in efforts of dengue prevention. The application includes a dengue symptom checker to match the various Dengue symptoms and recommends nearest clinics to the users. The application also includes an 'Input Location' function that allows users to monitor dengue clusters in other parts of Singapore other than the user's live location. Users will be notified if near an active dengue cluster.

Having such relevant information can help users to be more strategic in planning the locations to visit and take necessary actions in response to the information available.

1.5 References

Google Map API - https://developers.google.com/maps/

Android Studio- https://developer.android.com/studio/

Java SE 14.10.0 -
https://www.oracle.com/java/technologies/javase-downloads.html

CHAS Clinics-Data.gov.sg - https://data.gov.sg/dataset/chas-clinics

Dengue Clusters-Data.gov.sg - https://data.gov.sg/dataset/dengue-clusters

Visual Paradigm 16.1 - https://circle.visual-paradigm.com/docs/

2. Overall Description

2.1 Product Perspective

This product is a new, self-contained product to aid Singapore residents in fighting dengue. The software incorporates existing Google Maps API functions together with available dengue and clinic information from public government resources. Currently, the software can only be used within the premises of Singapore as the available data used is only limited to Singapore. However, this is subjected to portability.

The product aims to equip users with all the information they need to reduce the risk of getting dengue or a situation.

2.2 Product Functions

Main Functions

- Live Location
- Input Location
- Dengue Cluster Map
- Notification System
- Symptom Checker
- Nearest N Clinics Recommender

2.3 User Classes and Characteristics

Users could be from all age groups and need not have pre-knowledge for using the dengue preventive application. They want to be updated on the dengue cluster locations and check dengue symptoms with the MozzyShield application.

2.4 Operating Environment

The product will be built using Android Studios with Java programming language and requires integration of existing APIs like Google Maps. Our application can run on Android devices (preferably Samsung phones) with Android 10 and above.

2.5 Design and Implementation Constraints

2.5.1 The application can only run on Android devices with Android 10 and above.

2.5.2 The application will be developed using Android Studios with Java as the programming language.

2.5.3 This application will only be in English.

2.5.4 This application can only be used in Singapore at the moment.

2.5.5 New dengue cluster information is not updated live. Data files must be downloaded manually.

2.5.6 This application cannot run in the background.

2.5.7 This application works with Android devices of version 10 and above. (Samsung devices are preferred)

2.5.8 Known issues with Android Studio and Android Gradle Plugin - https://developer.android.com/studio/known-issues.

2.6 Assumptions and Dependencies

The product is programmed using Android Studios which is a free platform to develop applications for all developers.

3. External Interface Requirements

3.1 User Interfaces

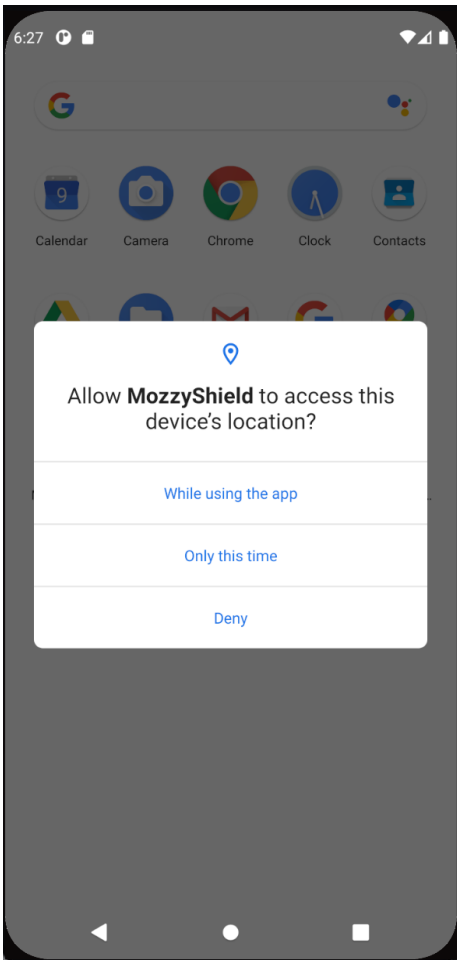

Figure 1: Request Location Permissions

Figure 2: User's Live Location

Figure 3: 'Clusters' Button

Figure 4: When 'Clusters' Markers are Clicked

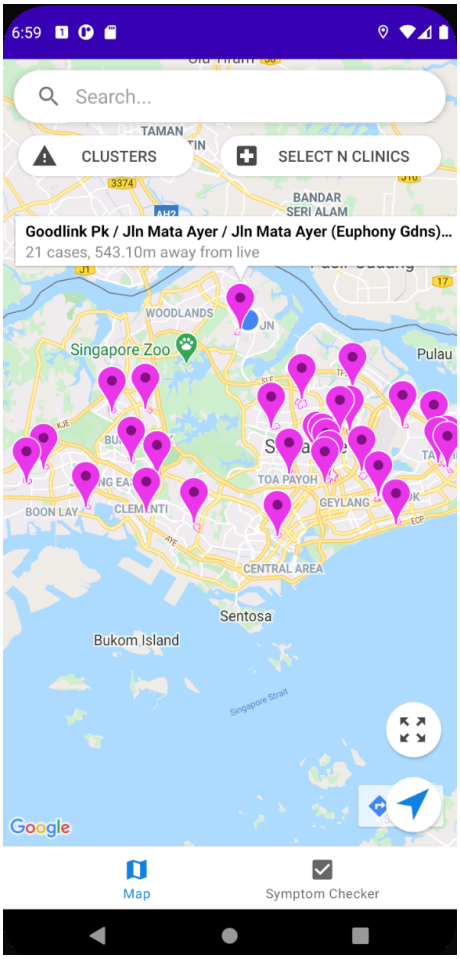

Figure 5: 'Overview' Button

Figure 6: 'Centralizer' Button

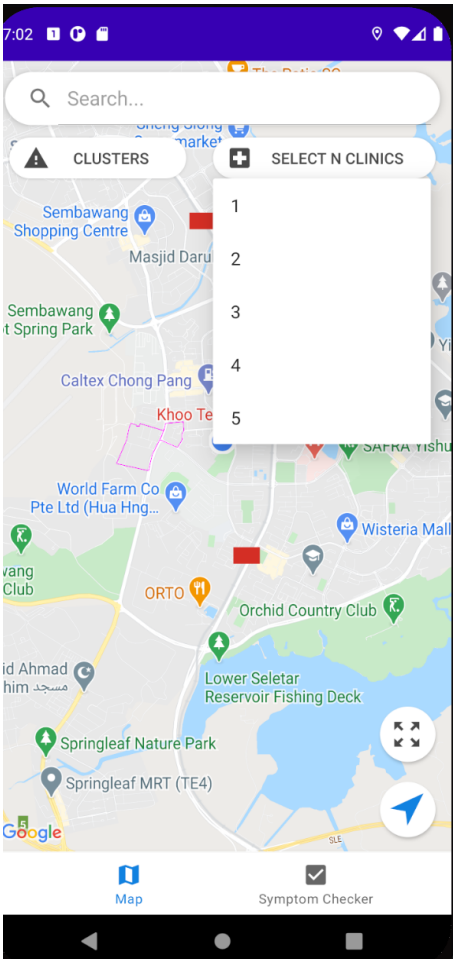

Figure 7: 'Select N Clinics' Button

Figure 8: User Inputs '5' Clinics

Figure 9: When 'Clinic' Markers are Clicked

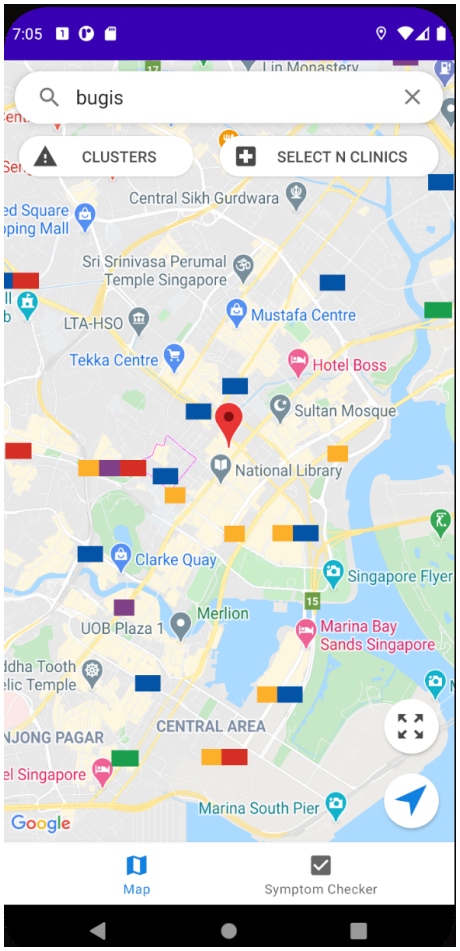

Figure 10: User Inputs 'Bugis' in 'Search' bar

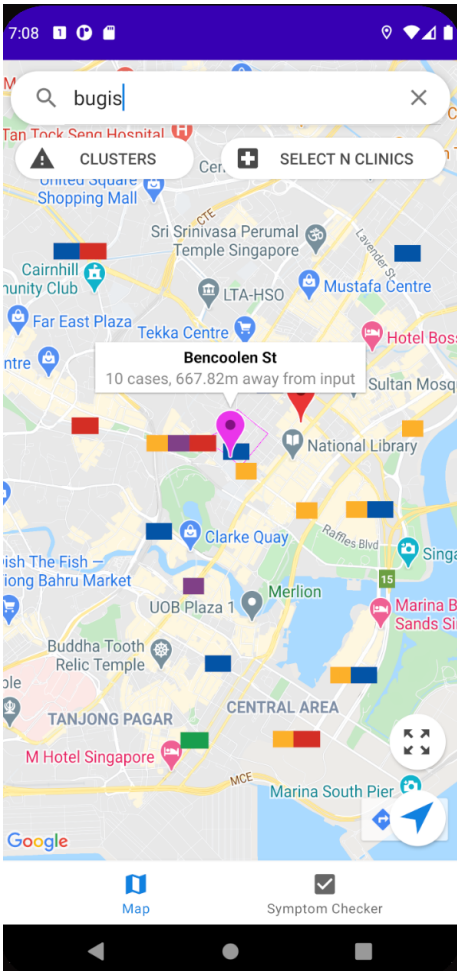

Figure 11: Cluster Marker relative to User's Input Location

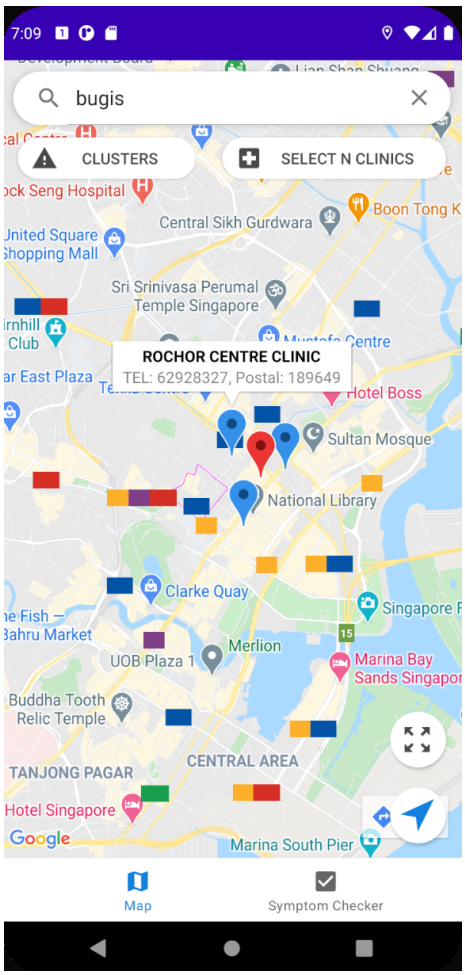

Figure 12: Clinic Marker relative to User's Input Location

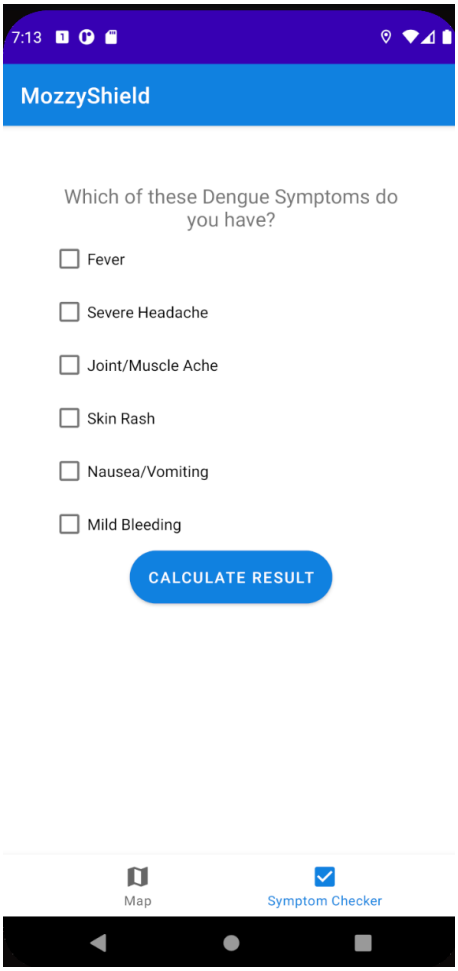

Figure 13: Symptom Checker

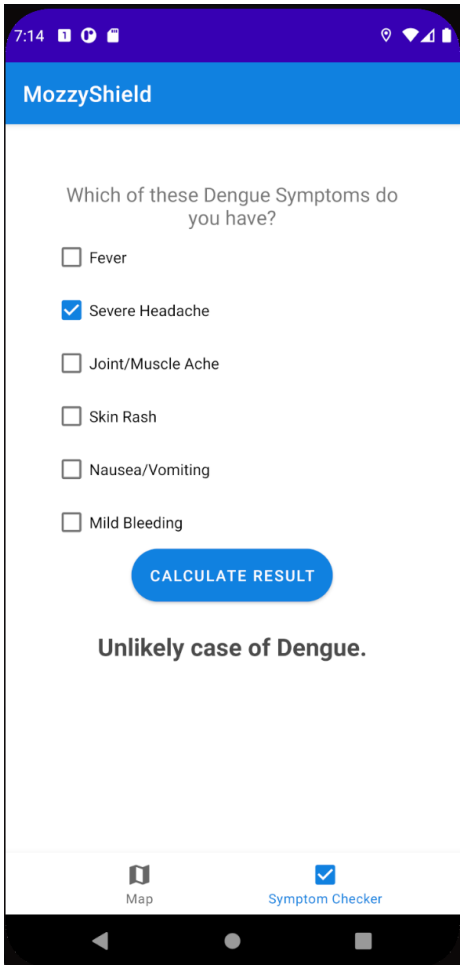

Figure 14: User ticks 'Severe Headache'

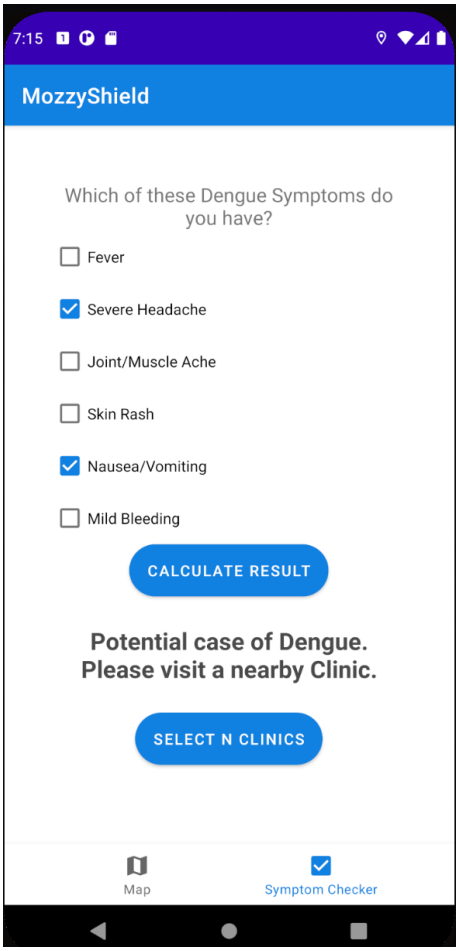

Figure 15: User ticks more Symptoms

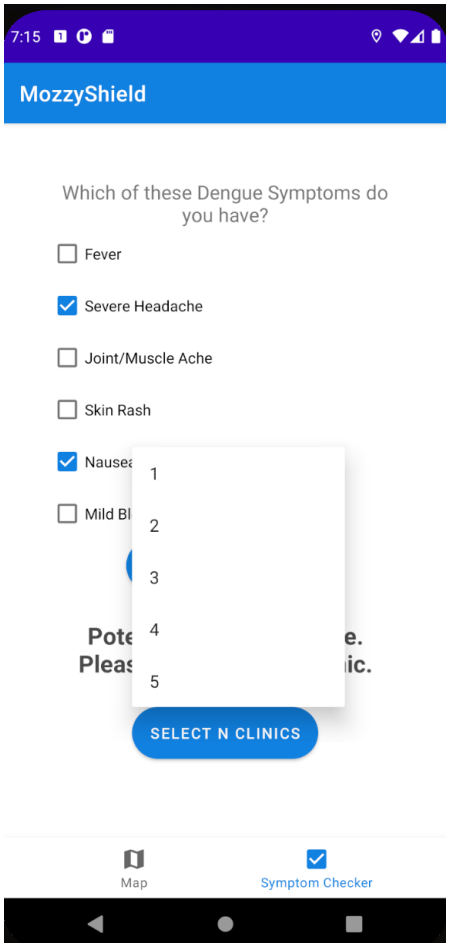

Figure 16: 'Select N Clinics' Button

Figure 17: User Inputs '4' Clinics

(UI Mockup - https://www.figma.com/file/MEnrFqL6cUmSNNyZp1QptO/UI-Mockup?node-id=94%3A12)

3.2 Hardware Interfaces

3.2.1 Only touch screen devices can be used to ensure proper functionality of the application.

3.2.2 The application must have GPS, Wifi and Mobile Data capabilities to ensure proper functionality of the application.

3.3 Software Interfaces

The application must run on Android devices (preferably Samsung devices) with Android 10 or above.

3.4 Communications Interfaces

3.4.1 The application must have access to push notifications, internet access and location permissions.

3.4.2 The application must be able to read in data files (GeoJson files) for Dengue Clusters & CHAS Clinics datasets.

4. Functional Requirements

4.1. Live Map

4.1.1. System must retrieve the dengue cluster information from Dengue Clusters-Dataset from data.gov.sg database.

4.1.2. Live location permission must be enabled.

4.1.3. ‘Live Location Marker’ (Blue Circle with White Border) and Dengue Cluster Borders highlighted in pink must always be shown on the map.

4.1.4. System must only display either ‘Nearest Clinic Markers’(Blue) or ‘Cluster Markers’ (Pink) at any point in time.

4.1.4.1. System must display ‘Live Location’ pop-up upon clicking on ‘Live Location Marker'.

4.1.5. User shall click on Buttons displayed on the Live Map.

4.1.5.1. User shall click on ‘Select N Clinics’ Button to display Blue markers.

4.1.5.2. User shall click on ‘Clusters’ Button to display Pink markers.

4.1.5.3. User shall click on ‘Overview’ Button.

4.1.5.4. User shall click on ‘Centralizer’ Button.

4.1.6. User shall click on ‘Search...’ to input location (Red).

4.2. User Input Location in ‘Search’ Bar

4.2.1. User shall enter postal code or name of input location.

4.2.2. System will display the ‘Input Location Marker’ on the Live Map with respect to their input location.

4.2.3. System must still display ‘Live Location Marker’ on the Live Map.

4.2.4. User shall click on ‘Clusters’ Button or ‘Select N Clinics’ Button with respect to their input location.

4.3. Send Notification when Near Dengue Clusters

4.3.1. System must retrieve the dengue cluster information from Dengue Clusters-Dataset from data.gov.sg database.

4.3.2. System will send a notification to inform the User of the dengue clusters within one kilometer of the User's 'Live Location Marker'.

4.3.3. System shall display the Live Map with the 'Cluster Markers' within one kilometer of the User's current location.

4.4. Symptom Checker

4.4.1. User clicks on the 'Symptom Checker' icon in the navigation bar at the bottom right of the screen.

4.4.2. System must display a list of dengue symptoms.

4.4.3. User shall indicate which symptoms they have by selecting the checkboxes.

4.4.4. System must calculate and output the likely percentage the User has Dengue.

4.4.5. System must also display the 'Select N Clinics' Button with an appropriate message when the percentage likelihood of Dengue is higher than or equals to 50%.

4.4.6. System must display an appropriate message when the percentage likelihood of Dengue is lower than 50%.

4.4.7. System must be able to load the Live Map after User clicks 'Select N Clinics'

4.5. Select N Clinics

4.5.1. User shall view the nearest N (1 to 5) clinics from either of the 2 interfaces as listed.

4.5.1.1. User can view nearest clinics from Live Location by clicking ‘Select N Clinics’ Button at the Live Map page or from Input Location after inputting location at the ‘Search’ Bar.

4.5.1.2. User can view nearest clinics from Live Location by clicking ‘Select N Clinics’ Button from the Symptom Checker page.

4.5.2. User shall indicate the number of clinics from 1 to 5 to view on the Live Map.

4.5.3. System must retrieve the clinic's information from CHAS Clinics data.gov.sg database.

4.5.4. System displays the clinic’s information as a pop up on each marker on click.

4.5.4.1. Pop up displays the name of the clinic.

4.5.4.2. Pop up displays the telephone number of the clinic.

4.5.4.3. Pop up displays the postal code of the clinic.

4.5.5. User shall click on ‘Centralizer’ Button to re-centre the Live Map to view the User’s ‘Live Location Marker’ and removes all other Markers on the map.

4.6. Display Clusters

4.6.1. User shall click on ’Cluster’ Button.

4.6.2. System must retrieve the dengue cluster information from Dengue Clusters-Dataset from data.gov.sg database.

4.6.2.1. System displays the cluster’s information as a pop up on each marker on click.

4.6.2.2. Pop up will show the name of the area of the dengue cluster.

4.6.2.3. Pop up will show the distance away of the dengue clusters from the User’s ‘Live Location Marker’ when there is no Input Location

4.6.2.4. Pop up will show the Pop up will show the distance away of the dengue clusters from the User’s ‘Input Location Marker’ when there is an Input Location

4.6.2.5. Pop up will show the number of dengue cases in the dengue clusters.

5. Non-functional Requirements

5.1. Performance

5.1.1. System should alert users and send notification within 30 seconds if the user is less than one kilometer away from a dengue cluster for necessary precautions to be taken immediately.

5.1.2. Results for the symptom checker should be generated within 5 seconds.

5.1.3. Pages must take less than 30 seconds to load their respective information.

5.1.4. The system must respond to a user input within 10 seconds.

5.1.5. Live Location must be up to date with at most 10 seconds delay after Live Location marker appears on map.

5.2. Security

5.2.1. User's personal information must not be requested.

5.2.2. User's location must not be accessible by other parties at all times.

5.3. Usability

5.3.1. Users should be able to learn how to use the software with ease within 5 minutes.

5.3.2. The application can be used by literate users of all ages.

5.3.3. The application must provide informative feedback.

5.3.3.1. The application should display accurate error messages when the user inputs an inexistent location in the search bar.

5.3.4. Rate of error by users is very low due to low requirements for user input that may cause error.

5.4. Reliability

5.4.1. After the application reboots, the functionality of the application must be restored within 15 seconds.

5.4.2. After the user allows the sharing of their live location, the application should access their location securely without being hacked or leaked.

5.4.3. The application should stop the location access once the user exits the application.

5.5. Supportability

5.5.1. The system should be functional in any touch screen device with Android operating system (Android 10 and above).

5.5.1.1. System works optimally on Samsung devices.

5.6. Extensibility

5.6.1. The system allows easy extension of new functions through our design considerations

5.6.1.1. Strategy Pattern allows easy implementation of new features in the future (Refer to Section 7.1.1.1)

5.6.1.2. Single Responsibility Principle (SRP) allows easy implementation of new features in the future (Refer to Section 7.1.2)

6. Analysis Models

6.1. Use Case Diagram

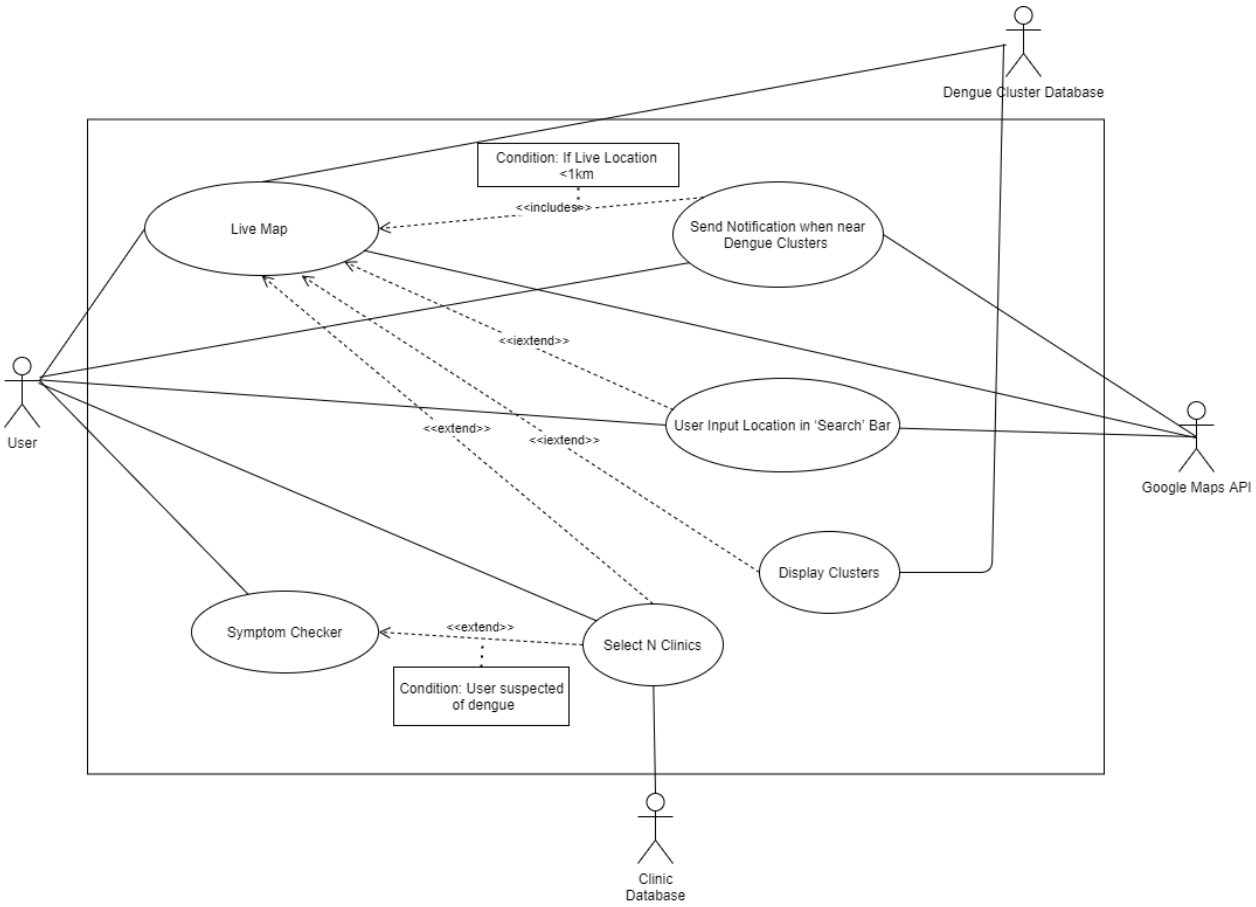

Figure 18: Use Case Diagram

6.2. Use Case Descriptions

6.2.1. Live Map

Use Case ID:	UC-1		
Use Case Name:	Live Map		
Created By:	Aloysius Seow	**Last Updated By:**	Qian Yi
Date Created:	29 January 2021	**Date Last Updated:**	9 April 2021

Actor:	User, Google Maps API, Dengue Clusters Database
Description:	Users must be able to view all Dengue Cluster Borders in Singapore, their Live Location, and click on 'Select N Clinics' Button, 'Clusters' Button, 'Overview' Button, 'Centralizer' Button and 'Search' Bar.
Preconditions:	● User must have Stable Internet Connection ● User must allow Live Location Permission
Postconditions:	Users shall be informed of ● Dengue Clusters Information ● Nearby N Clinics Information
Priority:	High
Frequency of Use:	Frequently
Flow of Events:	1. System fetches Dengue Cluster data from data.gov.sg data file. 2. System fetches Google Map data from Google Map API. 3. System displays the 'Live Location Marker' and Dengue Cluster Borders once the Map loads. 4. System will display all Buttons on the screen. 5. User shall click on 'Overview' Button to zoom out to view the entire map of Singapore.

	6. User shall click on ‘Centralizer’ Button to re-centre the Live Map to view the User’s ‘Live Location Marker’ and removes all other Markers on the map.
Alternative Flows:	AF-S5: If User clicks on ‘Clusters’ Button, 1. Go to ‘Display Clusters’ Use Case (UC-6) AF-S5: If User clicks on ‘Select N Clinics’ Button, 1. Go to ‘Select N Clinics’ Use Case (UC-5)
Exceptions:	-
Includes:	Send notification when near a Dengue clusters
Special Requirements:	-
Assumptions:	System must only display either ‘Nearest Clinic Markers’ or ‘Cluster Markers’ at any point in time. ‘Live Location Marker’ and Dengue Cluster Borders must always be shown on the map.
Notes and Issues:	-

6.2.2. User Input Location in 'Search' Bar

Use Case ID:	UC-2		
Use Case Name:	User Input Location in 'Search' Bar		
Created By:	Aloysius Seow	**Last Updated By:**	Qian Yi
Date Created:	29 January 2021	**Date Last Updated:**	9 April 2021

Actor:	User, Google Maps API
Description:	Users must be able to enter postal code or name of input location to show the input location on the Live Map, and shall use the 'Select N Clinics' Button or 'Clusters' Button to view information near the input location.
Preconditions:	● User must have Internet Connection ● User must allow Live Location Permission
Postconditions:	User will see the 'Input Location Marker' on the map and the information near the input location.
Priority:	Medium
Frequency of Use:	Occasionally
Flow of Events:	1. User shall enter postal code or name of input location. 2. System will display the 'Input Location Marker' on the Live Map with respect to their input location. 3. System must still displays 'Live Location Marker' on the Live Map. 4. User shall click on 'Clusters' Button or 'Select N Clinics' Button with respect to their input location.
Alternative Flows:	AF-S4: If User clicks on 'Clusters' Button, 1. Go to 'Display Clusters' Use Case (UC-6) AF-S4: If User clicks on 'Select N Clinics' Button,

	1. Go to ‘Select N Clinics’ Use Case (UC-5) AF-S4: If User clicks on ‘Centralizer’ Button, 1. System removes ‘Clinic Marker’ and ‘Cluster Markers’ and re-centers to User’s live location. AF-S4: If User clicks on ‘Overview’ Button, 1. System zoom out to view the entire map of Singapore.
Exceptions:	EX1: If User inputs a wrong format of postal code or locations with same names around the world 1. System catches the error and displays an error message.
Includes:	
Special Requirements:	-
Assumptions:	System must only display either ‘Nearest Clinic Markers’ or ‘Cluster Markers’ at any point in time. ‘Live Location Marker’ and Dengue Cluster Borders must always be shown on the map.
Notes and Issues:	-

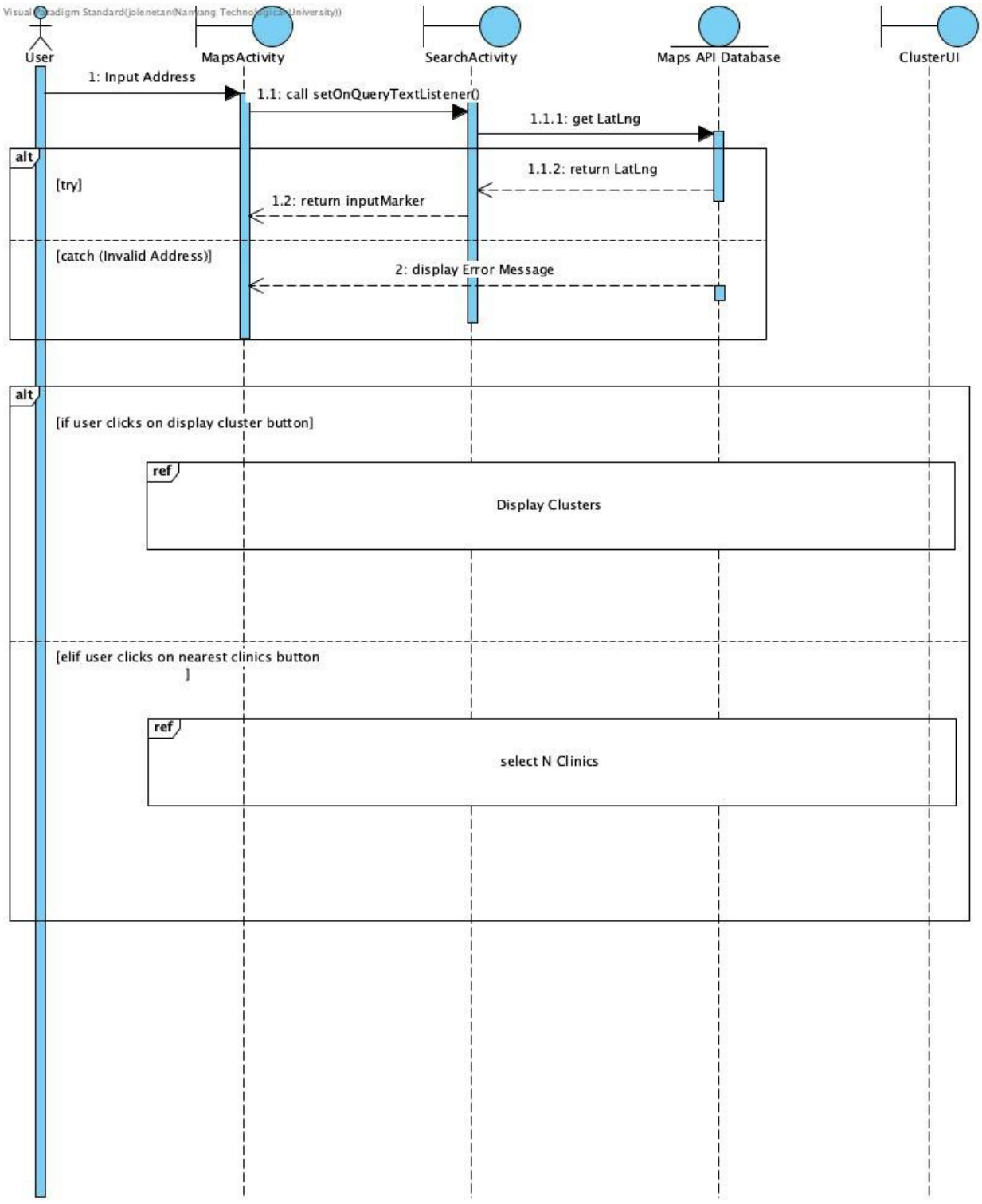

Figure 19: User Input Location in 'Search' Bar Sequence Diagram

6.2.3. Send Notification when Near Dengue Clusters

Use Case ID:	UC-3		
Use Case Name:	Send Notification when Near Dengue Clusters		
Created By:	Qian Yi	**Last Updated By:**	Qian Yi
Date Created:	7 February 2021	**Date Last Updated:**	9 April 2021

Actor:	User, Google Maps API, Dengue Clusters Database
Description:	Notification will be sent to the User when their Live Location is less than one kilometer away from a dengue cluster
Preconditions:	● User must have Internet Connection ● User must allow Live Location Permission ● User must be active in the application
Postconditions:	Dengue cluster statistics of the nearby dengue clusters will be displayed to the User
Priority:	Moderate
Frequency of Use:	Occasionally
Flow of Events:	1. System will send a notification to inform the User that they are less than one kilometer away from a nearby Dengue Cluster. 2. System shall display the 'Cluster Markers' on the Live Map.
Alternative Flows:	AF-S2: 1. User dismisses notification.
Exceptions:	-
Includes:	-
Special Requirements:	-

Assumptions:	User must be active in the application
Notes and Issues:	-

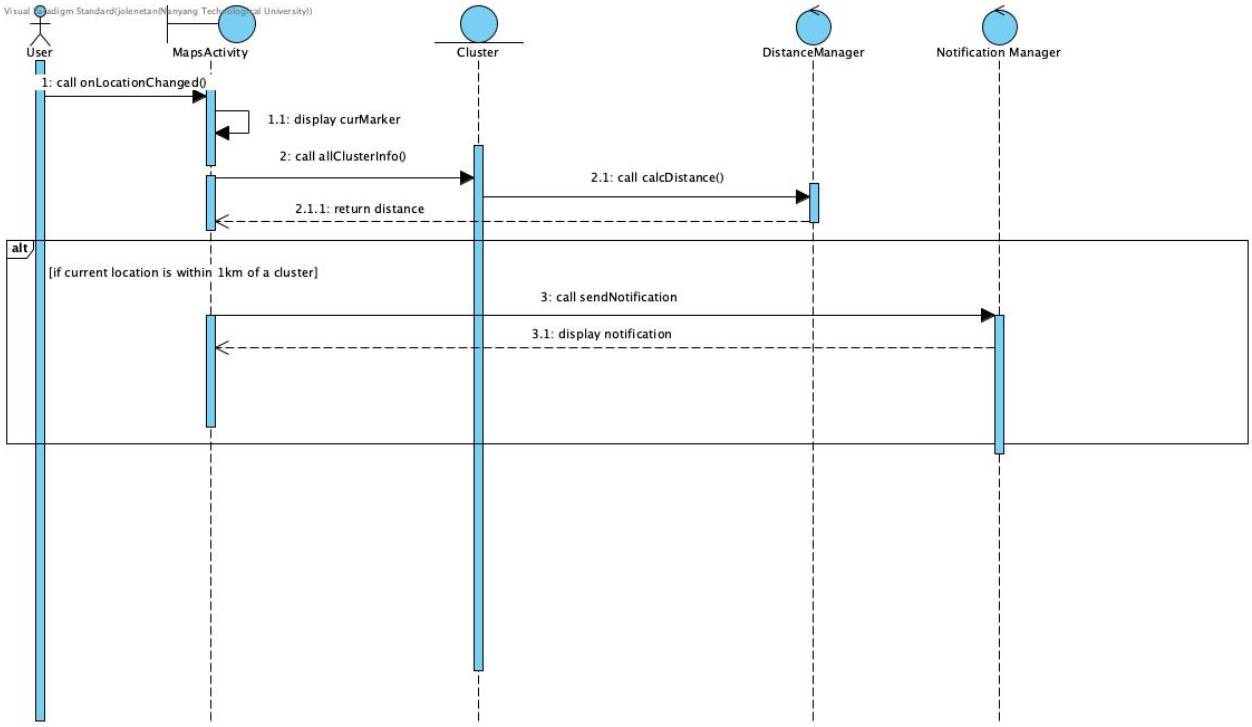

Figure 20: Send Notification when Near Dengue Clusters Sequence Diagram

6.2.4. Symptom Checker

Use Case ID:	UC-4		
Use Case Name:	Symptom Checker		
Created By:	Qian Yi	**Last Updated By:**	Qian Yi
Date Created:	30 January 2021	**Date Last Updated:**	9 April 2021

Actor:	User, Google Maps API
Description:	User inputs their symptoms by selecting checkboxes and the system will calculate the probability of the User having Dengue. System also suggests nearest clinics to User's live location.
Preconditions:	● User must have Internet Connection ● User must allow Live Location Permission
Postconditions:	Users have a sense of how likely they are to have Dengue, and the nearby clinics available.
Priority:	Medium
Frequency of Use:	Rarely
Flow of Events:	1. User clicks on the 'Symptom Checker' icon in the navigation bar below. 2. System displays a dengue symptom list. 3. User will indicate which symptoms they have by selecting the checkboxes. 4. System calculates and outputs the likely percentage the User has Dengue. 5. System will display the 'Select N Clinics' Button if the percentage likelihood of Dengue is higher than or equals to 50%.

Alternative Flows:	AF-S5: If the percentage likelihood of Dengue is lower than 50%: 1. System must display an appropriate message.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

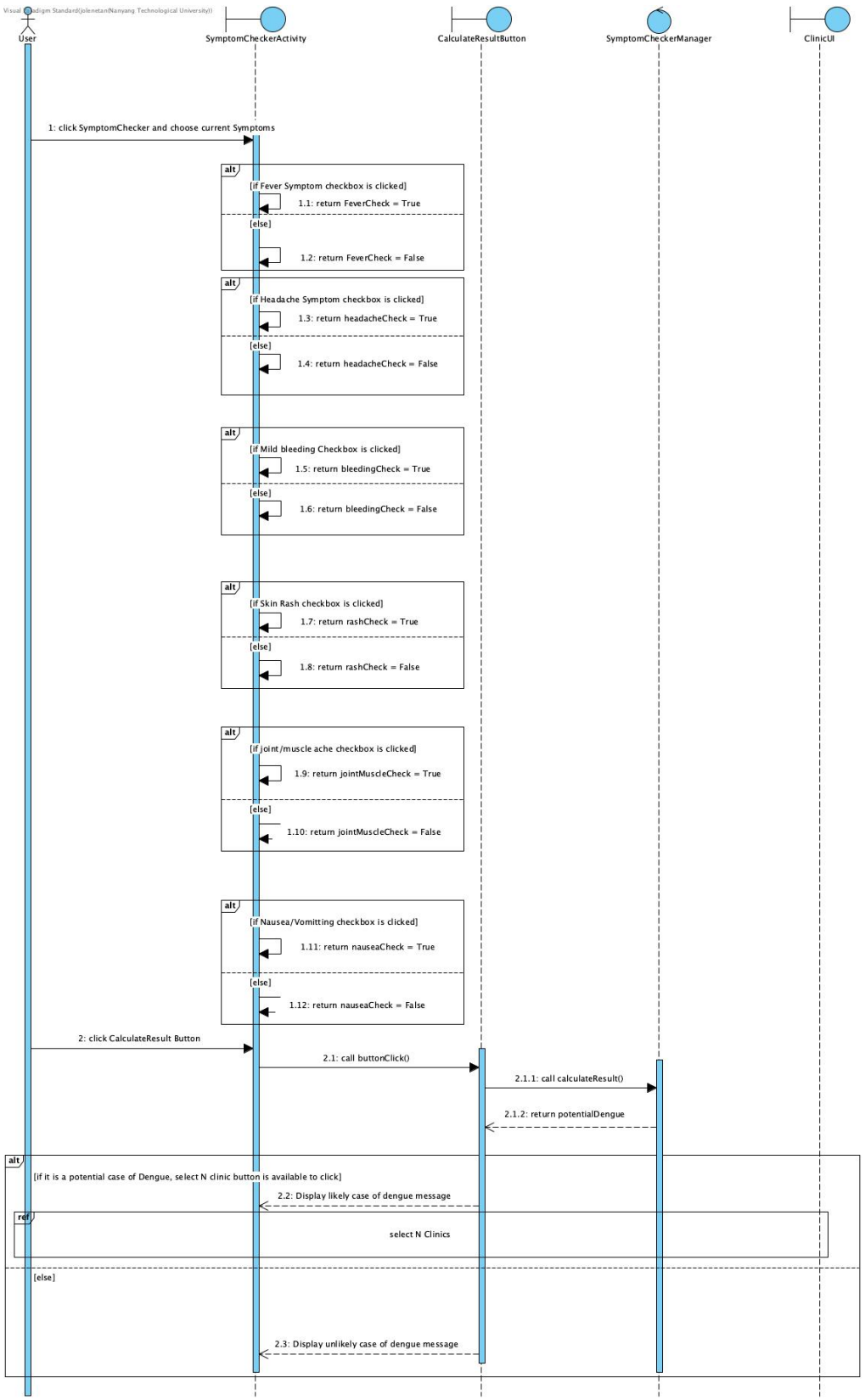

Figure 21: Symptom Checker Sequence Diagram

6.2.5. Select N Clinics

Use Case ID:	UC- 5		
Use Case Name:	Select N Clinics		
Created By:	Qian Yi	**Last Updated By:**	Qian Yi
Date Created:	7 February 2021	**Date Last Updated:**	9 April 2021

Actor:	User, Clinic Database, Google Maps API
Description:	Displays the nearest N clinics and its information to the User's live location.
Preconditions:	● User must have Internet Connection ● User must allow Live Location Permission
Postconditions:	Users view the information about the nearest N clinics to User's live location.
Priority:	Medium
Frequency of Use:	Rarely
Flow of Events:	1. User shall click the 'Select N Clinics' Button from Live Map. 2. User shall choose the number of clinics they want to see from 1 to 5. 3. System will display the 'Clinic Markers' information with respect to the User's 'Live Location Marker' marker if there is no Input Location. 4. System displays the clinic's information when User clicks on the 'Clinic Marker'. 5. User shall click on the Centralizer marker to remove the display of the clinic's information and markers.
Alternative Flows:	AF-S1: User shall click 'Select N Clinics' Button from Symptom Checker Page: 1. System will lead to the Live Map page.

	2. Go to MF-S4 AF-S3: System will display the ‘Clinic Markers’ information with respect to the User’s ‘Input Location Marker’ if User uses Input Location.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	System must only display either ‘Nearest Clinic Markers’ or ‘Cluster Markers’ at any point in time. ‘Live Location Marker’ and Dengue Cluster Borders must always be shown on the map.
Notes and Issues:	-

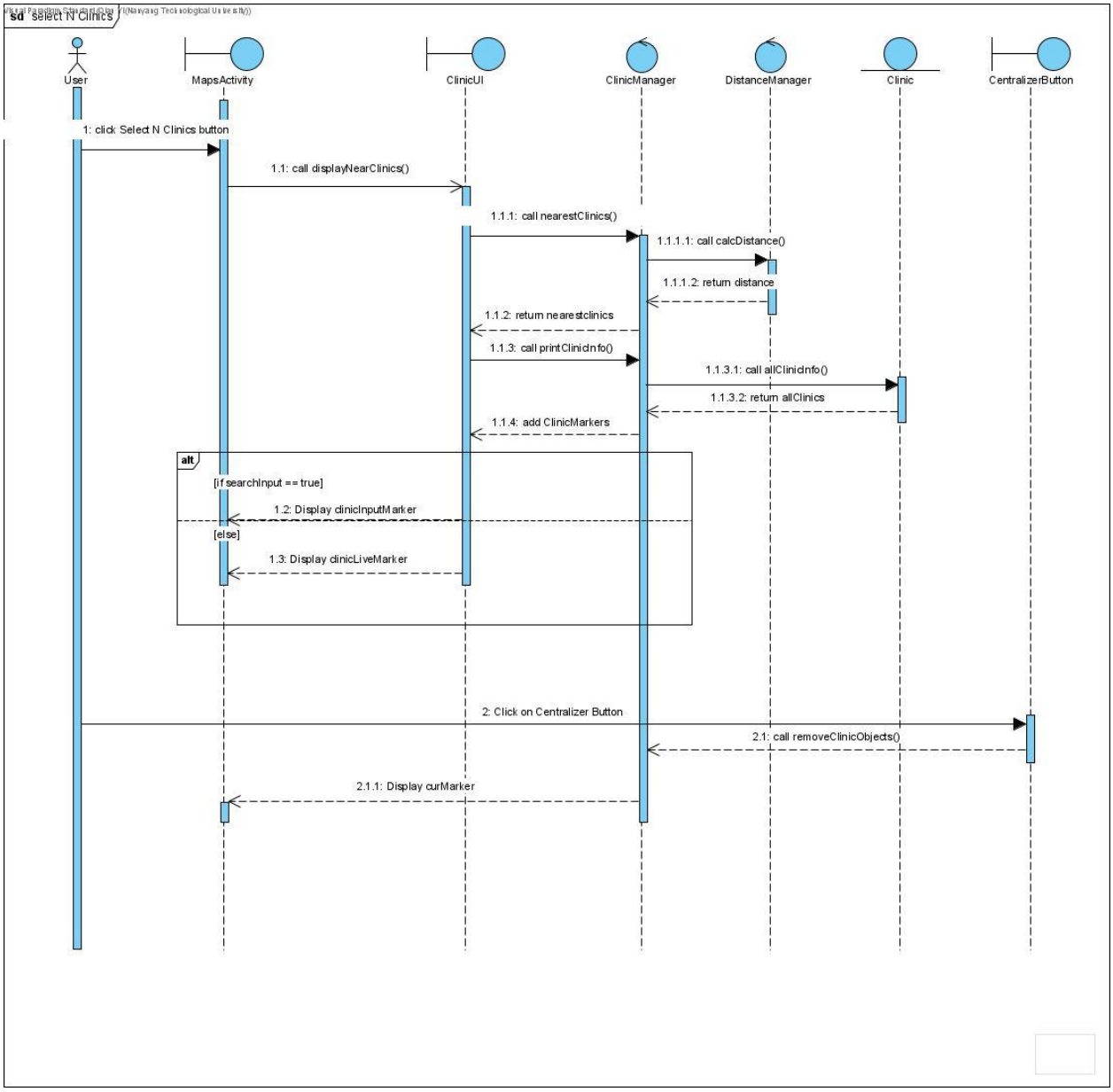

Figure 22: Select N Clinics Sequence Diagram

6.2.6. Display Clusters

Use Case ID:	UC-6		
Use Case Name:	Display Clusters		
Created By:	Qian Yi	**Last Updated By:**	Qian Yi
Date Created:	7 February 2021	**Date Last Updated:**	9 April 2021

Actor:	User, Google Maps API, Dengue Cluster Database
Description:	Displays Dengue Cluster and its information with respect to the User's 'Live Location' or 'Input Location'.
Preconditions:	● User must have Internet Connection ● User must allow Live Location Permission
Postconditions:	Users are able to view information about the Dengue Clusters.
Priority:	High
Frequency of Use:	Frequent
Flow of Events:	1. User shall click on the 'Cluster' Button. 2. System will retrieve the clusters' information from the Dengue Cluster database. 3. System will display a marker for each cluster on the User's Live map. 4. System displays the cluster's information as a pop up on each marker on click. 5. User shall click on the Centralizer marker to remove the display of the clinic's information and markers.
Alternative Flows:	-
Exceptions:	**-**
Includes:	**-**
Special Requirements:	**-**

Assumptions:	System must only display either ‘Nearest Clinic Markers’ or ‘Cluster Markers’ at any point in time. ‘Live Location Marker’ and Dengue Cluster Borders must always be shown on the map.
Notes and Issues:	-

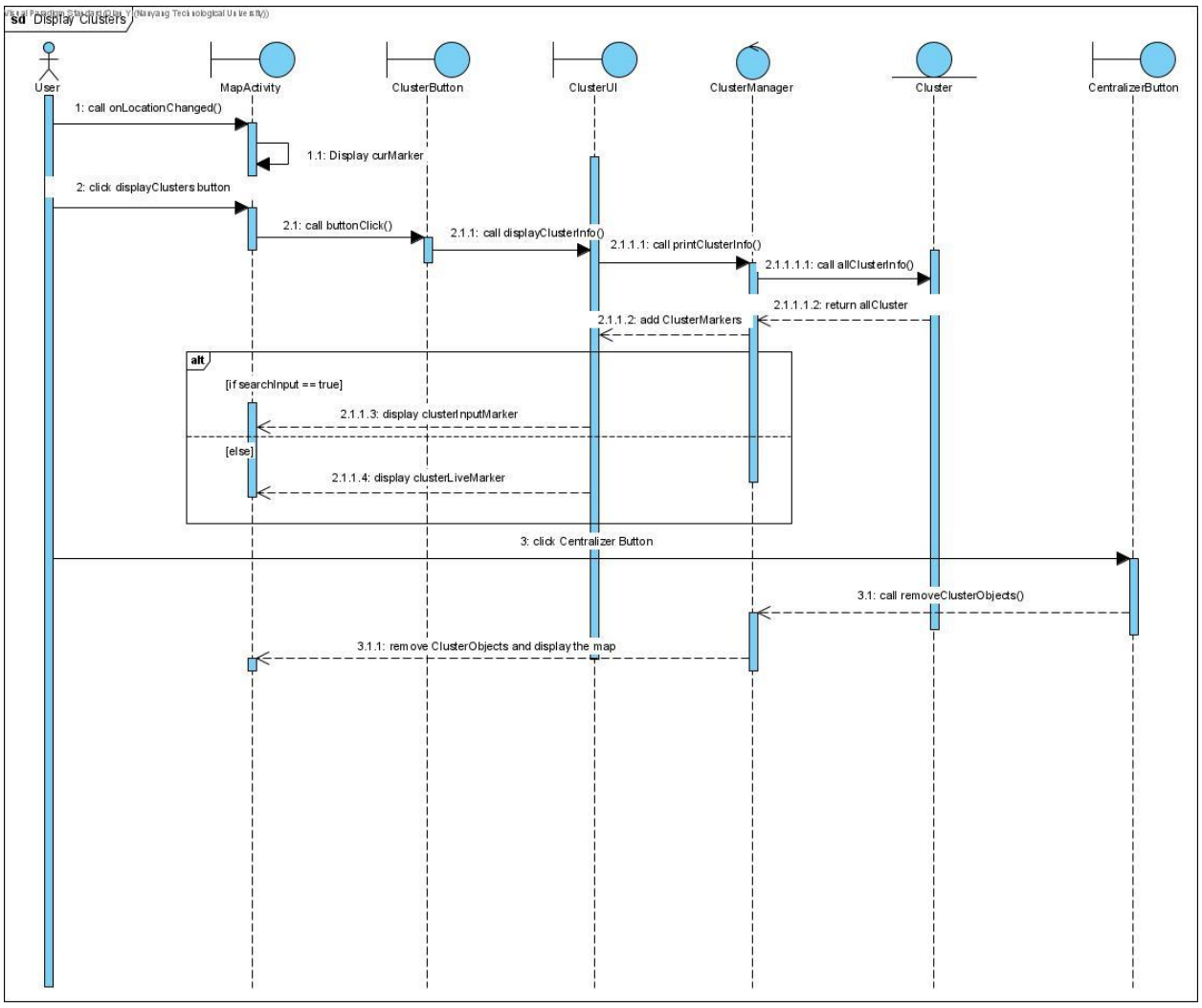

Figure 23: Display Clusters Sequence Diagram

6.3. Dialog Map

This dialog map describes the user interaction with our application and all the functions available.

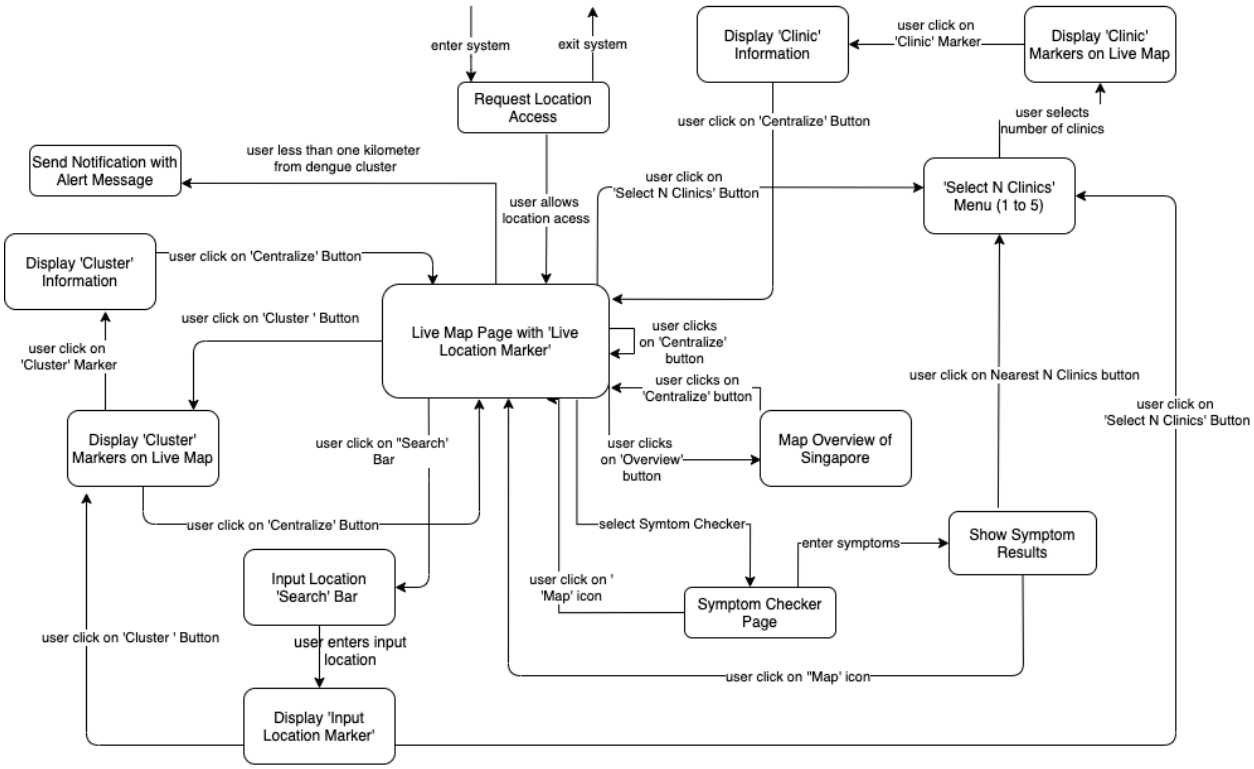

Figure 24: Dialog Map

6.4. Class Diagram

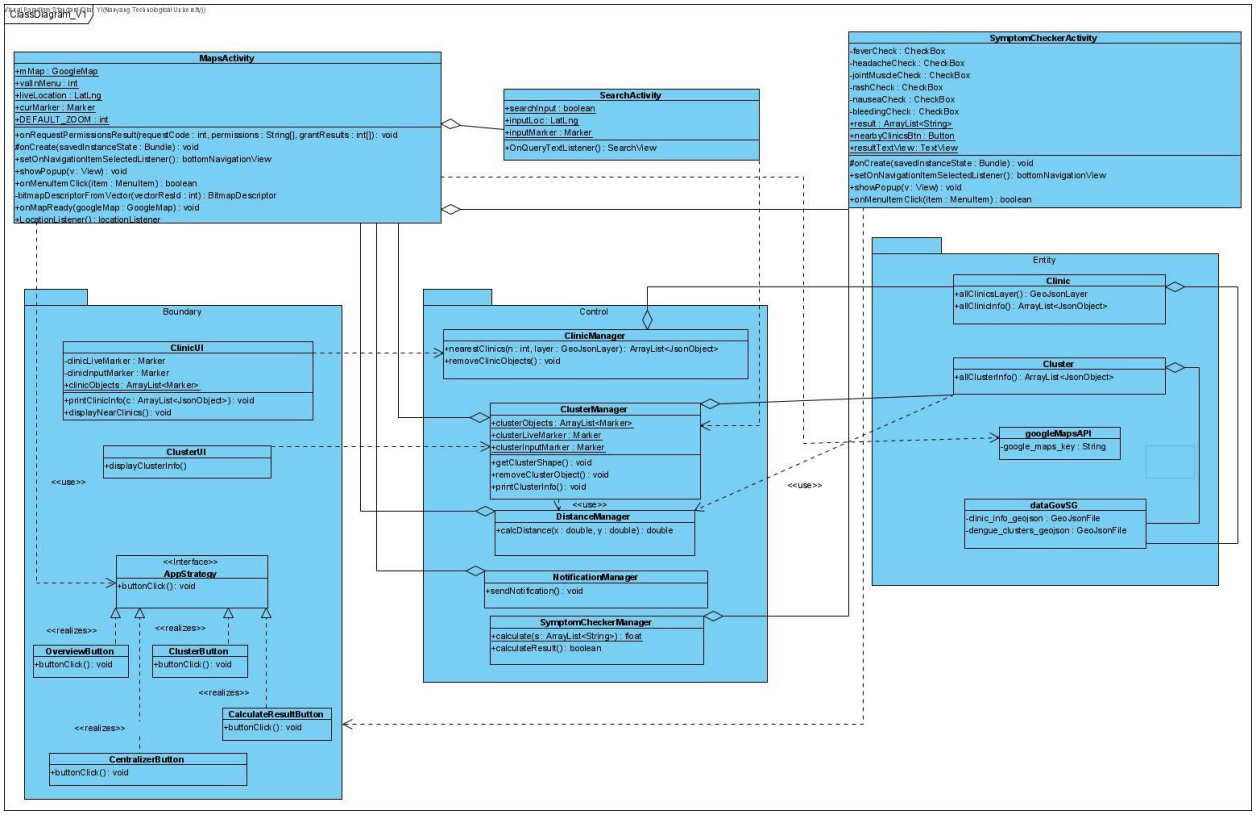

Figure 25: Class Diagram

6.5. Layer Architecture Diagram

The three-layered architecture of our product is divided into three parts, which are namely the Presentation, App Logic and Data Layer, and are kept separated to ensure low-coupling of the layers and each layer is easily modified by adding or removing classes within the layers. The layer architecture ensures the dependencies are one directional and therefore only the upper stream layers can make calls to the next downstream or same stream layers. Layered architecture is a style that also strictly follows the Single Responsibility Principle which we have also used in our system as explained in Section 7.1.2. The logical separation of layers in one machine-platform helps minimize the impact of changes to be contained within each layer.

Our application uses Android Studio IDE hence the unidirectional flow starts with the **Presentation layer** which consists of UIs, activities and fragments. This layer is also in charge of receiving user inputs such as search location inputs and user-click buttons. This layer then calls the **App Logic layer**, which consist of application logic classes to process the user inputs and calls the third layer, the **Persistent Data laye**r which contains the database GEOJSON files. Taking 'Display Clusters' as an example, the user interacts through the cluster button in *ClusterUI* class (Presentation layer) which then calls *ClusterManager* class (App Logic Layer) to retrieve the data of the dengue clusters in the *Cluster* class (Persistent Data Layer) to display back to the user the dengue cluster locations on the map (Refer to Figure 23).

Compared to N-tiered architecture, our application focuses on the grouping of related functionalities into distinct 3 layers stacked vertically on each other. Additionally the communication between layers is done directly within the same machine such as the GEOJSON data files in our Persistent Data layer can directly communicate with our App Logic layer classes and Presentation layer classes. Hence there is no need for the physical separation of layers in different servers or machines like in N-tiered architecture.

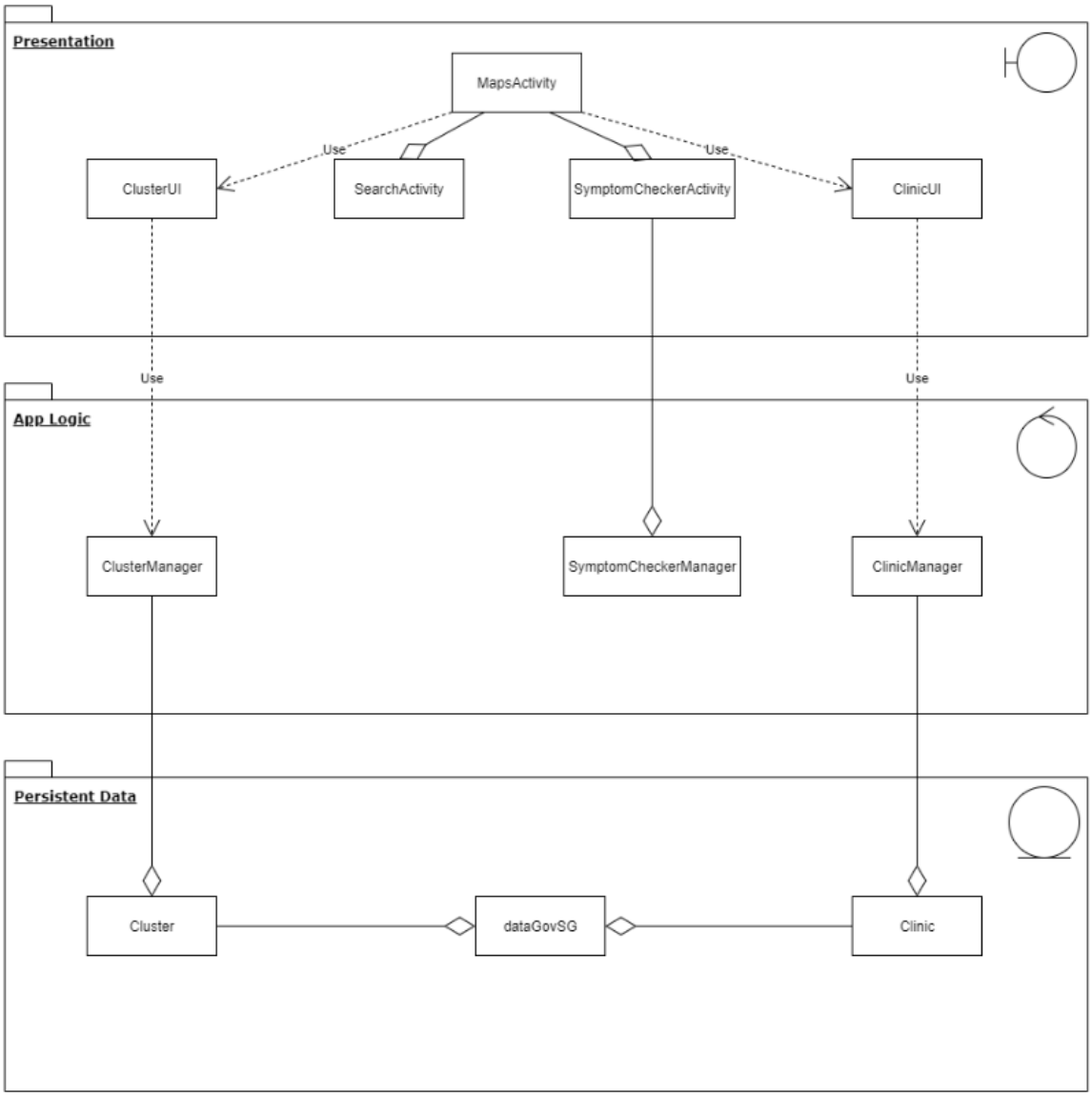

Figure 26: Layer Architecture Diagram

7. Design Patterns and Practices

7.1. Design Considerations

7.1.1. Design Problem

In our project, we noticed most of our user button-click functions can utilize the Strategy Design Pattern. Initially, all the functions of each button is implemented within one class itself which makes *MapsActivity* class (boundary class for the main map page) become tightly coupled to multiple repetitions of the button functions *(overviewOnClick, clusterButton, centralizerButton).* Additionally, the class does not become interchangeable as seen in the diagram below (Figure 27).

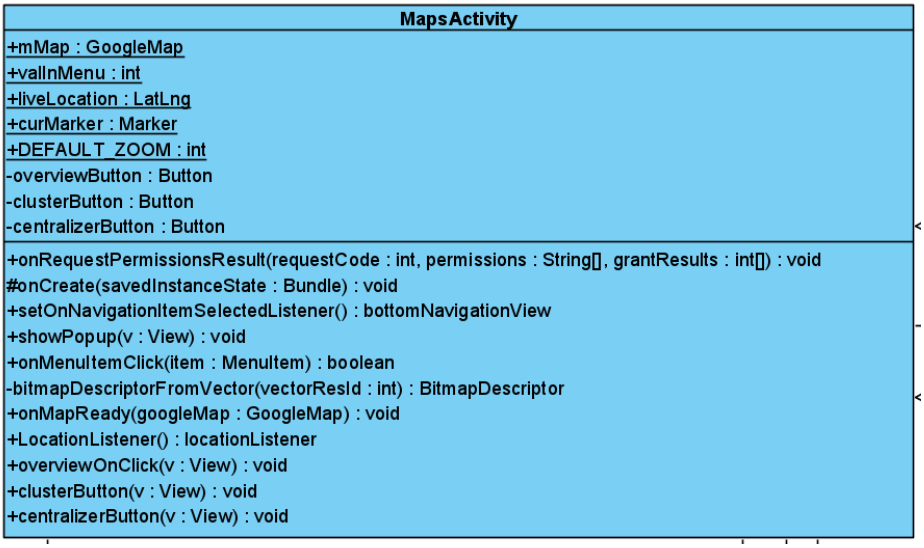

Figure 27: Extracted from Old Class Diagram

7.1.1.1. Strategy Pattern (Solution)

The Strategy Pattern encapsulates and hides the implementation information from other classes and the functions are only implemented upon specific user requests. By utilizing the Strategy Pattern, we added an interface class *(AppStrategy)* to allow the new button classes to be interchangeable with our contexts classes *(MapsActivity and SymptomCheckerActivity)*. This also allows our classes to also be open for extension for other forms of button functions.

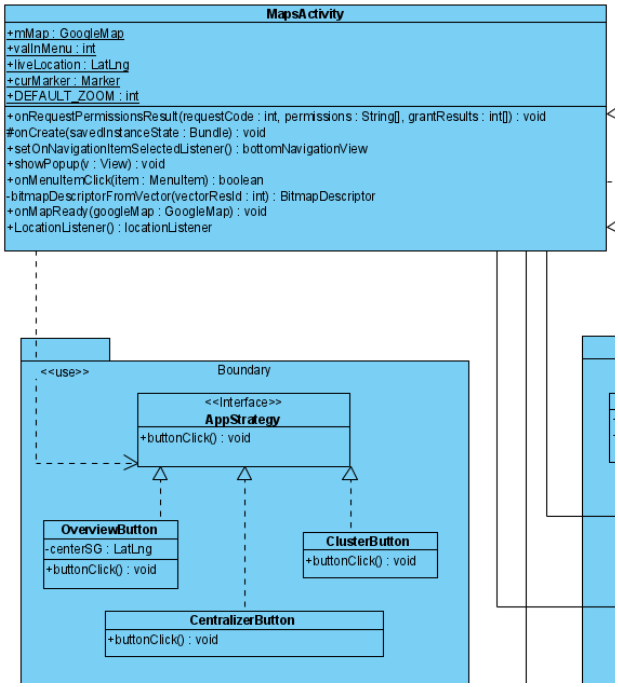

Figure 28: Strategy Pattern

7.1.2. Single Responsibility Principle (SRP)

The basis of SRP states that each responsibility is an axis of change. If a class assumes more than one responsibility, there will be more than one reason for it to change. While designing our application, we adhered to SRP and made each class responsible for only one feature of the application. This ensures that any change to each class will not affect another class, which also allows our application to be open to extension.

This is seen in our Class Diagram (*Figure 25*), one of our functions is to display all dengue clusters in Singapore on the map, and initially the *MapsActivity* class contained all the boundary, control and entity functions for the dengue cluster buttons (boundary), logic for displaying dengue clusters and markers (control) and dengue cluster GEOJSON files (entity). Using the SRP, we have split the function to 3 different classes - boundary class *clusterUI,* control class *clusterManager* and entity class *Cluster,* all contained respectively in each of the boundary, control and entity packages which follows the layered architecture diagram (Figure 26) in Section 6.5. This process is repeated for our other functions such as displaying the nearest (N) number of clinics in Singapore.

7.1.3. MVC Architecture

The MVC Architecture splits the system into the 3 separate components: **Model** contains the application data and business logic such as *Cluster* class in the entity package, **View** contains the UI logic for everything visible on the screen and user interaction such as *ClusterUI* class (boundary package, *xml* files). **Controller** is the glue between view and model and reacts to the user's input and presents the data requested by the user such as our *ClusterManager* class. The model component has no knowledge of the interface.

MVC has more emphasis on the View Layer and thus MVC allows for multiple Views for each Model which makes the UIs more open and flexible to changes, reusable and extensible. Our project utilizes this such as adding more UI features to the *Boundary* package (View Layer) does not require additional changes to the *Entity* package (Model layer). This ensures maintainability, and reusability of the components as modifications can be easily carried out in each MVC layer individually without affecting the other layers, allowing our application to be more open and flexible to changs.

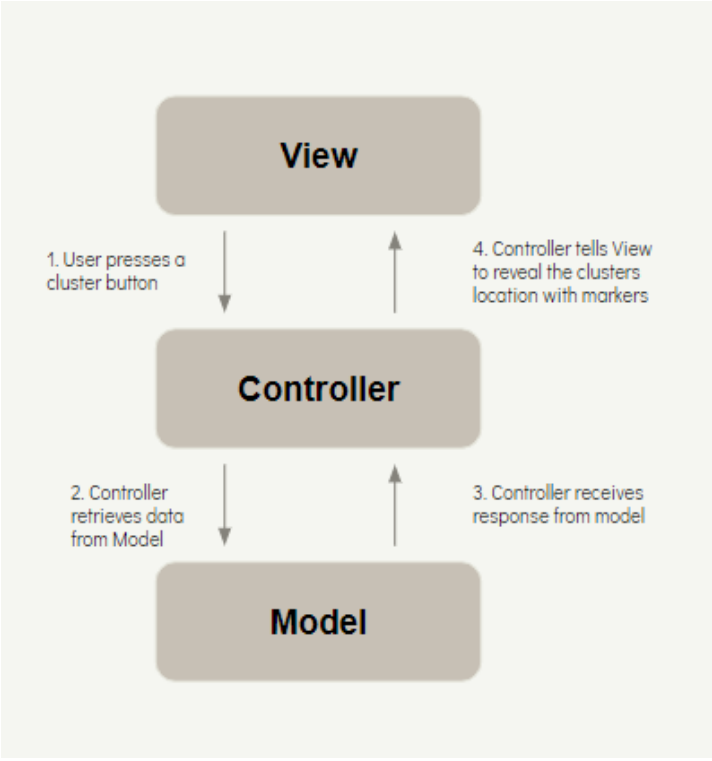

Figure 29: MVC Architecture Diagram

8. Testing

8.1. Blackbox Testing (Equivalence Class Testing)

8.1.1. Input Search Location

Name	Location	Input	Oracle	Log
Input Search Location	SearchActivity	637658	Hall of Residence 13, NTU	Hall of Residence 13, NTU
		880011	System Message: "Location not found, please enter another nearby location"	System Message: "Location not found, please enter another nearby location"

8.1.2. Notification within 1km of Dengue Cluster

Name	Location	Input	Oracle	Log
Notification within 1km of Dengue Cluster	NotificationManager	500 meters away from cluster	Notification shown on emulator	Notification shown on emulator
		1200 meters away from cluster	No notification shown on emulator	No notification shown on emulator

8.2. Whitebox Testing

8.2.1. Cluster UI

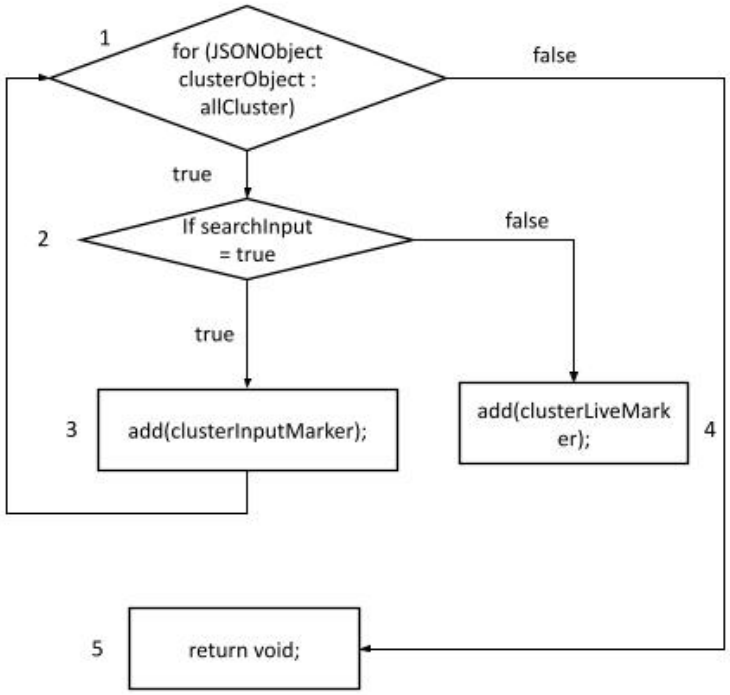

Figure 30: Cluster UI Whitebox Testing

Cyclomatic Complexity = |Edges| - |Nodes| + 2 = 2

Basis Path	Input	Expected Result	Actual Result
1, 2, 3, 1	637658	Add Cluster Markers with respect to Input Location marker on Live map	Add Cluster Markers with respect to Input Location marker on Live map
1, 2, 4	No Input	Add Cluster Markers with respect to Live Location marker on Live map	Add Cluster Markers with respect to Live Location marker on Live map

8.2.2. Navigation Bar (MapsActivity - onNavigationItemSelected())

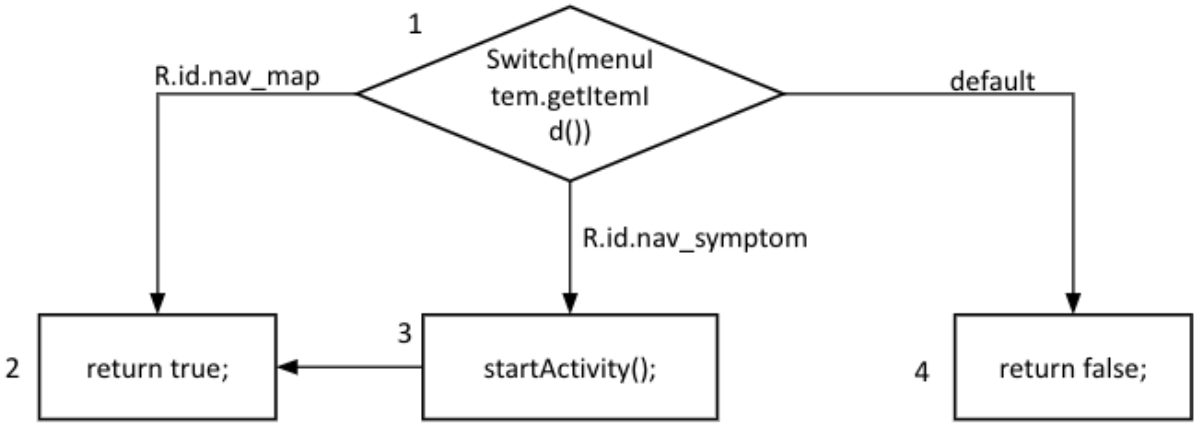

Figure 31: Navigation Bar Whitebox Testing

Cyclomatic Complexity = |Edges| - |Nodes| + 2 = 2

Basis Path	Input	Expected Result	Actual Result
1, 2	Click Map	Show Live Map page	Show Live Map page
1, 3, 2	Click Symptom Checker	Show Symptom Checker page	Show Symptom Checker page

8.2.3. ‘Select N Clinic’ Menu Button (onMenuItemClick())

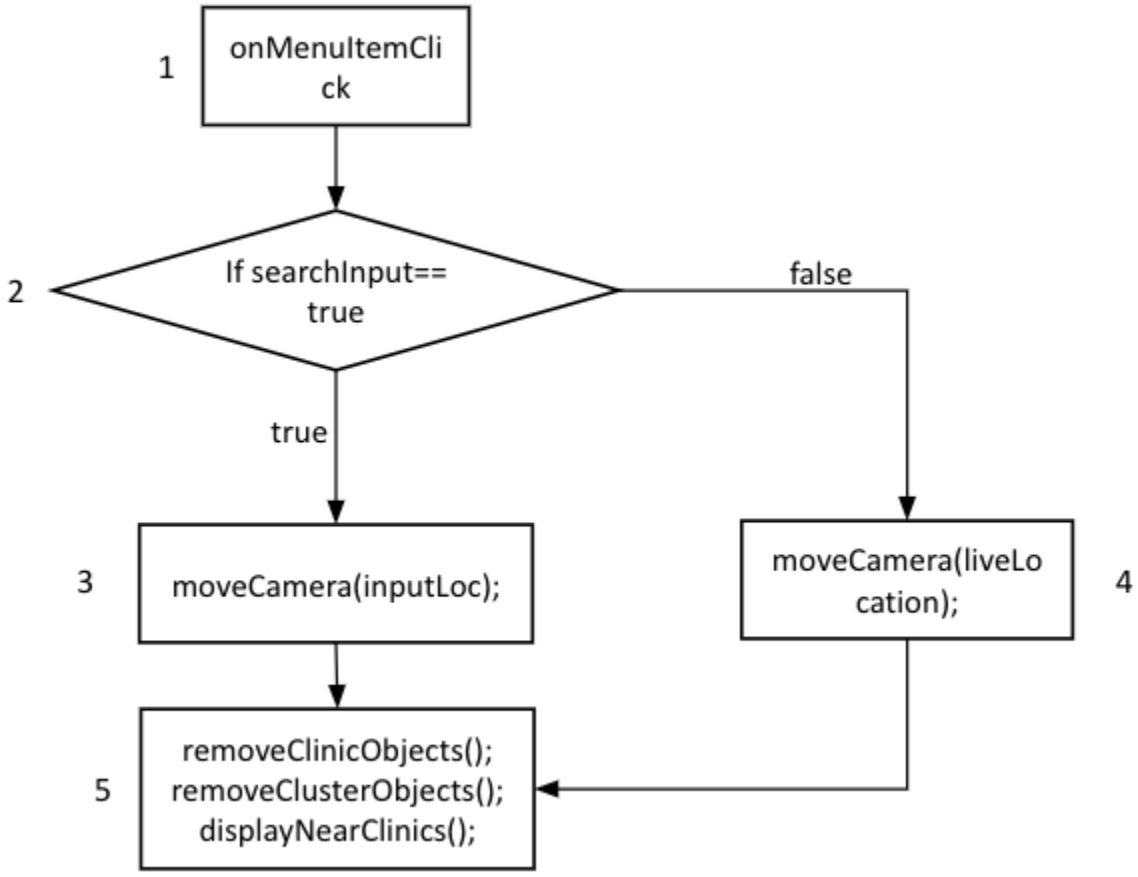

Figure 32: ‘Select N Clinic’ Menu Button Whitebox Testing

Cyclomatic Complexity = |Decision Points| + 1 = 2

Basis Path	Input	Expected Result	Actual Result
1, 2, 3, 5	searchInput==true	Add Clinic Markers with respect to Input Location marker on Live map	Add Clinic Markers with respect to Input Location marker on Live map
1, 2, 4, 5	searchInput==false	Add Clinic Markers with respect to Live Location marker on Live map	Add Clinic Markers with respect to Live Location marker on Live map

8.2.4. Symptom Checker (For 'Mild Bleeding' symptom - bleedingCheck.setOnClickListener())

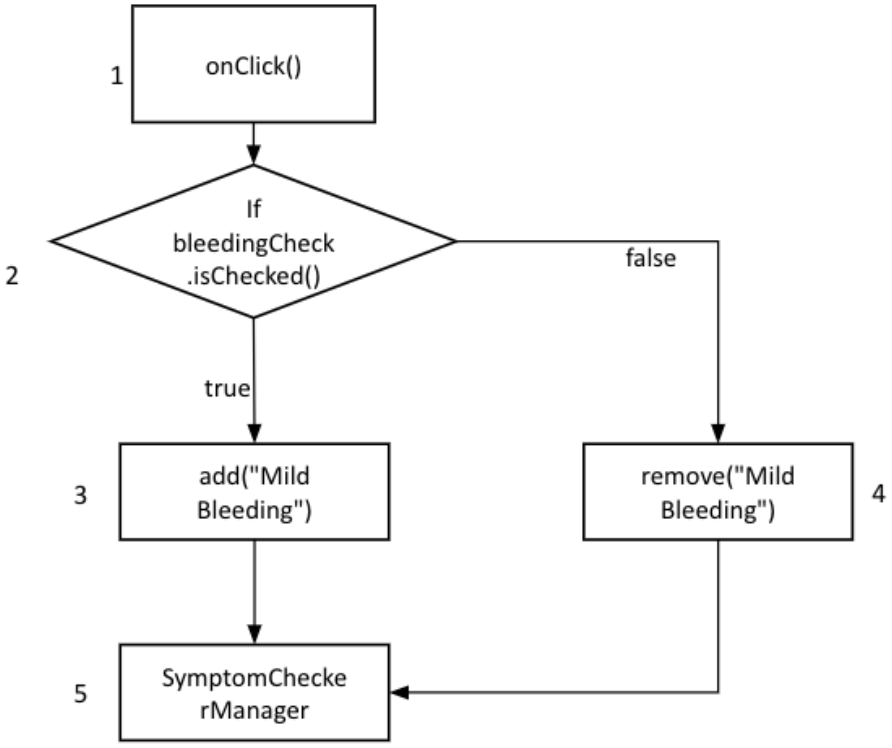

Figure 33: Symptom Checker Whitebox Testing

Cyclomatic Complexity = |Decision Points| + 1 = 2

Basis Path	Input	Expected Result	Actual Result
1, 2, 3, 5	Click "Mild Bleeding"	Check "Mild Bleeding"	Check "Mild Bleeding"
1, 2, 4, 5	Unclick "Mild Bleeding"	Uncheck "Mild Bleeding"	Uncheck "Mild Bleeding"

8.2.5. Show Percentage Likelihood of Dengue (calculateResultBtn.setOnClickListener())

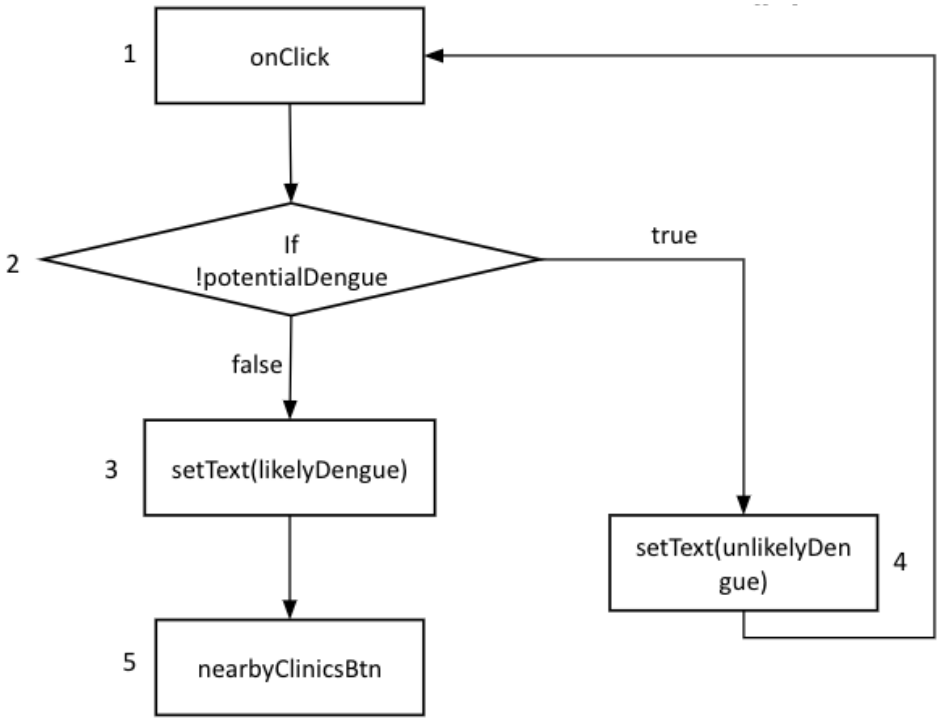

Figure 34: Show Percentage Likelihood of Dengue Whitebox Testing

Cyclomatic Complexity = |Decision Points| + 1 = 2

Basis Path	Input	Expected Result	Actual Result
1, 2, 3, 5	Dengue potential >= 50	Show text "Potential case of Dengue. Please visit a nearby Clinic."	Show text "Potential case of Dengue. Please visit a nearby Clinic."
1, 2, 4, 1	Dengue potential < 50	Show text "Unlikely case of Dengue."	Show text "Unlikely case of Dengue."

8.2.6. Calculate Total Percentage

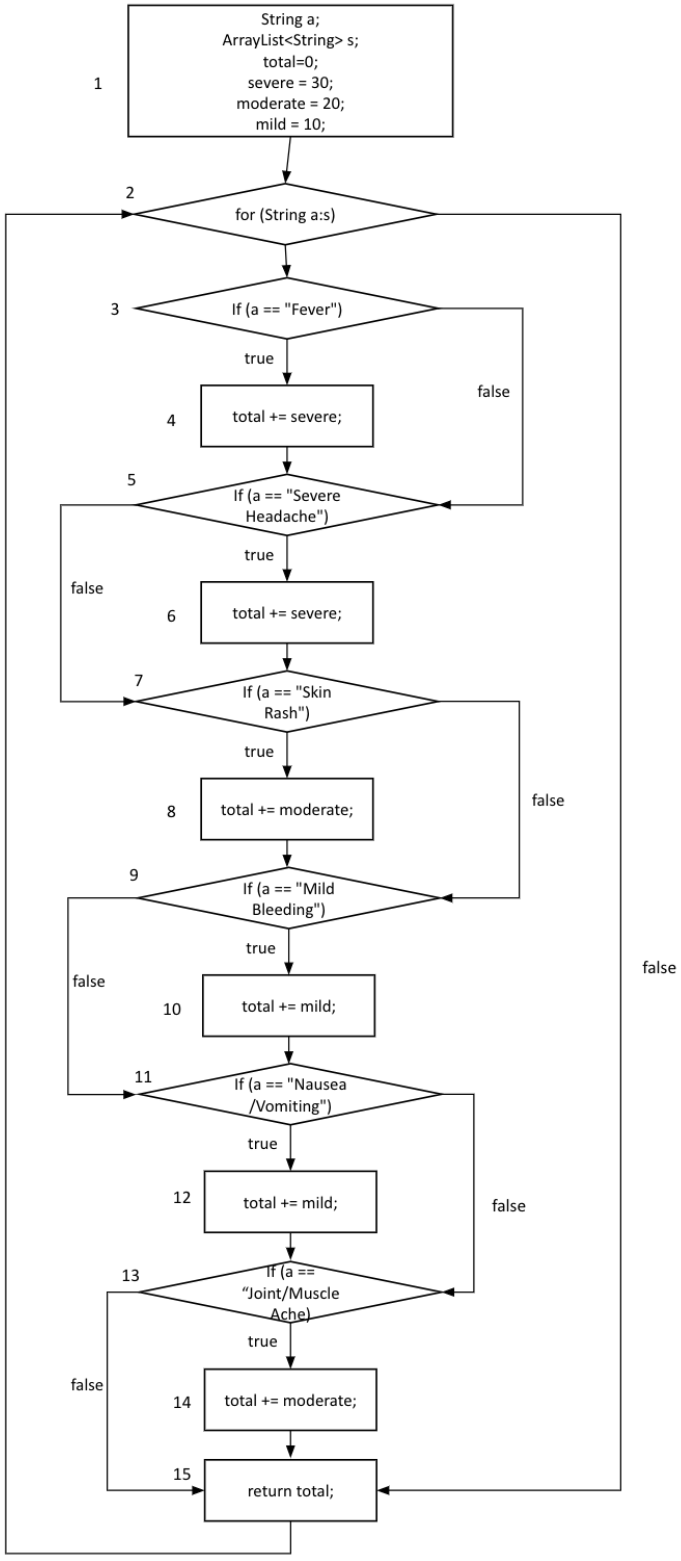

Figure 35: Calculate Total Percentage Whitebox Testing

Cyclomatic Complexity = |Decision Points| + 1 = 7

Basis Path	Input	Expected Result	Actual Result
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1	Fever, Severe Headache, Skin Rash, Mild Bleeding, Nausea /Vomiting, Joint/Muscle Ache	Total = 120	Total = 120
1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1	Severe Headache, Skin Rash, Mild Bleeding, Nausea /Vomiting, Joint/Muscle Ache	Total = 90	Total = 90
1, 2, 3, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1	Fever, Skin Rash, Mild Bleeding, Nausea /Vomiting, Joint/Muscle Ache	Total = 90	Total = 90
1, 2, 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 1	Fever, Severe Headache, Mild Bleeding, Nausea /Vomiting, Joint/Muscle Ache	Total = 100	Total = 100
1, 2, 3, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 1	Fever, Severe Headache, Skin Rash, Nausea /Vomiting, Joint/Muscle Ache	Total = 110	Total = 110
1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 1	Fever, Severe Headache, Skin Rash, Mild Bleeding, Joint/Muscle Ache	Total = 110	Total = 110
1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 1	Fever, Severe Headache, Skin Rash, Mild Bleeding, Nausea /Vomiting	Total = 100	Total = 100

Glossary

Term	Description
Dengue	An acute infectious disease caused by a flavivirus transmitted by female aedes mosquitoes, and characterized by headache, severe joint pain, and a rash.
Clusters	Clusters indicates a locality with active transmission where intervention is targeted. It is formed when two or more cases have onset within 14 days and are located within 150m of each other
Symptom Checker	A checklist for users to check their symptoms and for the application to check for the possibility of having dengue.
Clinics	Any clinic registered in the government's database under CHAS (Community Health Assist Scheme) excluding hospitals.
Live Location	Live location of the user's device that is running the application based on Global Position System (GPS)
Input Location	User's desired input location that is entered in the search bar (Postal Code/Name of Location).
Marker	A marker identifies a location on the map.
Pop-up	A pop-up displays statistics of a cluster/clinic when a marker is clicked.
Statistics of a Cluster	Statistics of a cluster may include Cluster's name, Number of Dengue Cases, and distance from Live Location/Input Location.

Statistics of a Clinic	Statistics of a clinic may include Clinic's name, Telephone Number of Clinic, and distance from Live Location/Input Location.
Notification	Refers to the device notification, not the in-application notification