CZ2006 Software Engineering
AY20/21 Sem 1

Final Documentation

By Team TYYM

Table of Content

1. PROJECT MISSION STATEMENT

The Teenage Yummy Yellow Men will develop a website to enhance ease of access to residents living in Singapore about the information of dengue cases in a chosen location. This project reinforces the Singapore National Environmental Agency's national dengue prevention campaign and raises safety awareness among Singapore residents.

2.1 FUNCTIONAL REQUIREMENTS

System Functionality to be performed

1. A user should be able to view the number of new Dengue cases reported in Singapore for that day.
2. A user should be able to view the number of dengue cases in their chosen location in the last 14 days.
3. A user should be able to view the past 7-day trend of the number of dengue cases in their chosen location.
4. A user should be able to view dengue prevention resources, compiled into one page for convenient reference.
5. A user should be able to report a dengue breeding ground to the NEA by sending an email through our app.

Interface with other systems

1. The system must be able to use wifi/data to communicate all transactions with data.gov.sg API.
2. The system must be able to use wifi/data to send emails to Contact_NEA@nea.gov.sg
3. The system must be able to be run on mobe (iOS, Android) systems.
4. Data must be pulled from the data.gov.sg API at a minimum frequency of once every 24 hours.

2.2 NON-FUNCTIONAL REQUIREMENTS

Usability

1. 90% of users must be able to obtain the number of dengue cases in their chosen area within 1 minute of starting to use the system.
2. The ‘View Cases’ page must contain a drop-down menu that provides options to users on the location they input.
3. The dashboard must have a minimalistic design, featuring a maximum of four pieces of information displayed on the page at any point in time.

Performance

1. When the user inputs a location, the system must generate the required output within 5 seconds.

3. DATA DICTIONARY

Term	Definition
data.gov.sg API	An application programming interface maintained by the Singapore Government that provides local public data
Dengue	Dengue virus is an Aedes mosquito-borne virus that results in dengue fever. So far, all 5 known serotypes of the virus are able to cause the full spectrum of the disease.
NEA	The National Environment Agency is a statutory board responsible for improving and sustaining a clean and green environment in Singapore
Number of cases	Number of patients confirmed to have dengue fever
Dengue Clusters	A dengue cluster is a localised area where two or more dengue cases have been recorded in less than 14 days
Red Alert Level	High-risk area with 10 or more cases
Yellow Alert Level	High-risk area with less than 10 cases
Green Alert Level	Low-risk area, under surveillance for the next 21 days

4. USE CASE DESCRIPTIONS

Use Case ID:	1.1		
Use Case Name:	Check Dengue Cases in Singapore for That Day		
Created By:	Xing Xiang	Last Updated By:	-
Date Created:	05/09/20	Date Last Updated:	-

Actor:	User, database
Description:	The country-wide number of dengue cases for the current day is shown to the User.
Preconditions:	1. The app boots successfully 2. The user's smartphone has internet connection
Postconditions:	1. The app successfully shows the country-wide number of cases for the current day to the user
Priority:	High
Frequency of Use:	1-2 times per day

Flow of Events:	1. The user opens the app 2. The app loads and presents the user with an input bar to input a specific location 3. The user selects the location from a dropdown box 4. The app queries the current day's island-wide case count from the database 5. The app presents the case count to the user
Alternative Flows:	AF-S4: If the database has not been updated from nea.gov.sg for the current day 1. The app uses the most recent set of data for step 5. 2. The app displays the message "This data is correct as of <insert date>"
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	1.2		
Use Case Name:	Check Dengue Cases in a Specified Location		
Created By:	Xing Xiang	Last Updated By:	Xing Xiang
Date Created:	28/08/20	Date Last Updated:	12/09/20

Actor:	User, Database
Description:	The user must select a specific location and is presented with a dashboard with the most recent 14-day total number of dengue cases, as well as the trend in the number of dengue cases for the past week. The interface will be color coded based on the derived dengue alert level.
Preconditions:	3. The app boots successfully 4. The user chooses the option to select a location 5. The user's smartphone has internet connection
Postconditions:	2. The app successfully shows the current 14-day-total cases to the user 3. The app interface is coloured in the correct dengue alert colour code
Priority:	High

Frequency of Use:	1-2 times per day
Flow of Events:	6. The user opens the app 7. The app loads and presents the user with an input bar to input a specific location 8. The user selects the location from a dropdown box 9. The app receives the location and sends a request to the Database API for all 14-day-total dengue cases in Singapore 10. The app filters the number of dengue cases in the selected location 11. The app presents the current 14-day-total cases to the user
Alternative Flows:	AF-S4: If the database has not been updated from data.gov.sg API 3. The app uses the previous set of data for steps 4-6. 4. The app displays the message "This data is correct as of <insert date>"
Exceptions:	EX1: The app loses internet connection while querying the API 1. The app displays "Oh no! We seem to have lost connection" 2. The app retries every 2 seconds to query the API once internet connection is back
Includes:	1.1: Check Dengue Cases in Singapore for That Day 1.3: Select Location 3.1: Show Trends 4.1: Derive Dengue Alert Colour Code
Special Requirements:	-

Assumptions:	-
Notes and Issues:	-

Use Case ID:	1.3		
Use Case Name:	Select Location		
Created By:	Xing Xiang	Last Updated By:	-
Date Created:	28/8/20	Date Last Updated:	-

Actor:	User
Description:	The user selects their location from a dropdown box.
Preconditions:	1. The “Select Location” dropdown box loads successfully 2. Permissions to get the user’s current location has been granted
Postconditions:	1. The location entered is saved and used for subsequent actions
Priority:	High
Frequency of Use:	1-2 times per day

Flow of Events:	1. The user taps the dropdown box 2. The app presents the user with a dropdown list of various locations, including a “Current Location” option at the top 3. The user selects a location
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	2.1		
Use Case Name:	Report Potential Breeding Spot		
Created By:	Darren	Last Updated By:	Darren
Date Created:	05/09/20	Date Last Updated:	12/09/20

Actor:	User, Email System
Description:	The user is able to report an area that he/she thinks is a potential breeding spot. After information about this potential breeding spot is provided by the user, it is then sent out to an external email system, and an email will be sent to the NEA to report this potential breeding spot
Preconditions:	1. The app boots successfully 2. The user selects the option to report potential breeding spot 3. The user's smartphone has internet connection
Postconditions:	1. The app successfully displays an acknowledgement message to the user
Priority:	High
Frequency of Use:	1-2 times per day

Flow of Events:	1. The user opens the app 2. The app loads and presents the user with 2 options: select a location, or report potential breeding spots 3. The user selects the option to report potential breeding spots 4. The user is then prompted by the system to fill up a form with information about the potential breeding site. Details about the form are provided in use case 2.2 5. The user is provided with a non-mandatory option to upload a photo of the potential breeding site 6. The app displays a message that acknowledges that the details have been successfully received, and an email will be sent to the NEA
Alternative Flows:	AF-S5: If the user chooses not the upload a photo of the potential breeding site 1. Step 5 is skipped
Exceptions:	EX1: The app loses internet connection while connecting to the external email system 1. The app displays “The information has been recorded, but an email is unable to be sent out now due to technical difficulties. The potential breeding spot will be reported to the NEA once these issues are resolved. ” 2. The app retries every 2 seconds to connect to the external email system once the internet connection is back
Includes:	2.1: Fill up form
Special Requirements:	-
Assumptions:	-

Notes and Issues:	-

Use Case ID:	2.2		
Use Case Name:	Fill Up Form		
Created By:	Darren	Last Updated By:	Darren
Date Created:	05/09/20	Date Last Updated:	12/09/20

Actor:	User
Description:	The user is required to fill up a form with details about the potential breeding spot. All of the input fields are situated within a single-page form, with only some fields being mandatory
Preconditions:	1. The app boots successfully 2. The user selects the option to report potential breeding spot 3. The user's smartphone has internet connection
Postconditions:	1. The app moves on to use case 2.3
Priority:	High
Frequency of Use:	1-2 times per day

Flow of Events:	1. The user is requested to input the date and time of occurrence (required field) 2. The user inputs street name (required field) 3. The user inputs building name (optional field) 4. The user inputs block or house number (optional field) 5. The user inputs the floor number (optional field) 6. The user inputs the unit number (optional field) 7. The user inputs the postal code (optional field) 8. The user inputs a description of the potential breeding site, capped at 5000 characters (required field) 9. The user clicks on the 'submit' button 10. The app saves the information provided by the user
Alternative Flows:	
Exceptions:	EX1: The user does not fill up all required fields 1. The app displays an error message "Please fill up all required fields before proceeding" 2. The app displays the same form again, with the user's previous inputs already filled
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	2.3		
Use Case Name:	Upload Image		
Created By:	Darren	Last Updated By:	
Date Created:	05/09/20	Date Last Updated:	

Actor:	User
Description:	The user is able to upload an image of the potential breeding spot
Preconditions:	1. The app boots successfully 2. The user selects the option to report potential breeding spot 3. The user has filled up the form (Use case 2.2 has been executed successfully) 4. The user's smartphone has internet connection
Postconditions:	5. Control is passed back to Use case 2.1
Priority:	High
Frequency of Use:	1-2 times per day

Flow of Events:	1. The user is presented with an option to upload an image or to take a photo 2. The app asks for the user's permission to access his/her camera 3. The user takes a photo of the potential breeding site 4. The user clicks on the 'upload' button 5. The app displays an acknowledgement message: "Your photo has been successfully uploaded."
Alternative Flows:	AF-S2: If the user chooses to upload a photo already in their library instead of taking a photo 2. The app asks for the user's permission to access his/her photo library 3. The user chooses a photo to upload
Exceptions:	EX1: The size of the image uploaded exceeds 3MB 1. The app displays "The size of the uploaded image is too large. Please upload another image with file size less than 3MB." 2. Steps 1-5 are repeated
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	3.1		
Use Case Name:	Show Trends		
Created By:	Dylan	Last Updated By:	-
Date Created:	05/09/20	Date Last Updated:	-

Actor:	Database
Description:	The user is able to observe trends of dengue cases of the past 14 days from a bar chart
Preconditions:	1. The app boots successfully 2. The user has selected a specific location 3. The user's smartphone has internet connection
Postconditions:	1. A bar chart detailing the number of dengue cases each day for the past 14 days will be displayed
Priority:	High
Frequency of Use:	1-2 times per day

Flow of Events:	1. The user is presented with the option to view trends at the specified input location 2. The app queries the database and retrieves the stored data of dengue cases in the past 14 days 3. The app displays the data in a bar chart form on the interface
Alternative Flows:	AF-S4: If the database has not been updated from data.gov.sg API 1. The app uses the previous set of data for steps 2-3. 2. The app displays the message “This data is correct as of <insert date>”
Exceptions:	EX1: The specified location does not have a data set of the past 14 days 1. The app displays “Dengue case trend not available for this location. Please input a nearby location.” 2. Steps 1-3 are repeated
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	4.1		
Use Case Name:	Derive Dengue Alert Colour Code		
Created By:	Dylan	Last Updated By:	
Date Created:	05/09/20	Date Last Updated:	

Actor:	Database
Description:	The app will determine the respective dengue alert colour code based on the number of dengue cases in the location. The display interface will imitate the colour of the dengue alert colour code.
Preconditions:	1. The app boots successfully 2. The user has selected a specific location 3. The user's smartphone has internet connection
Postconditions:	1. The interface display colour changes to the dengue alert colour
Priority:	High
Frequency of Use:	1-2 times per day

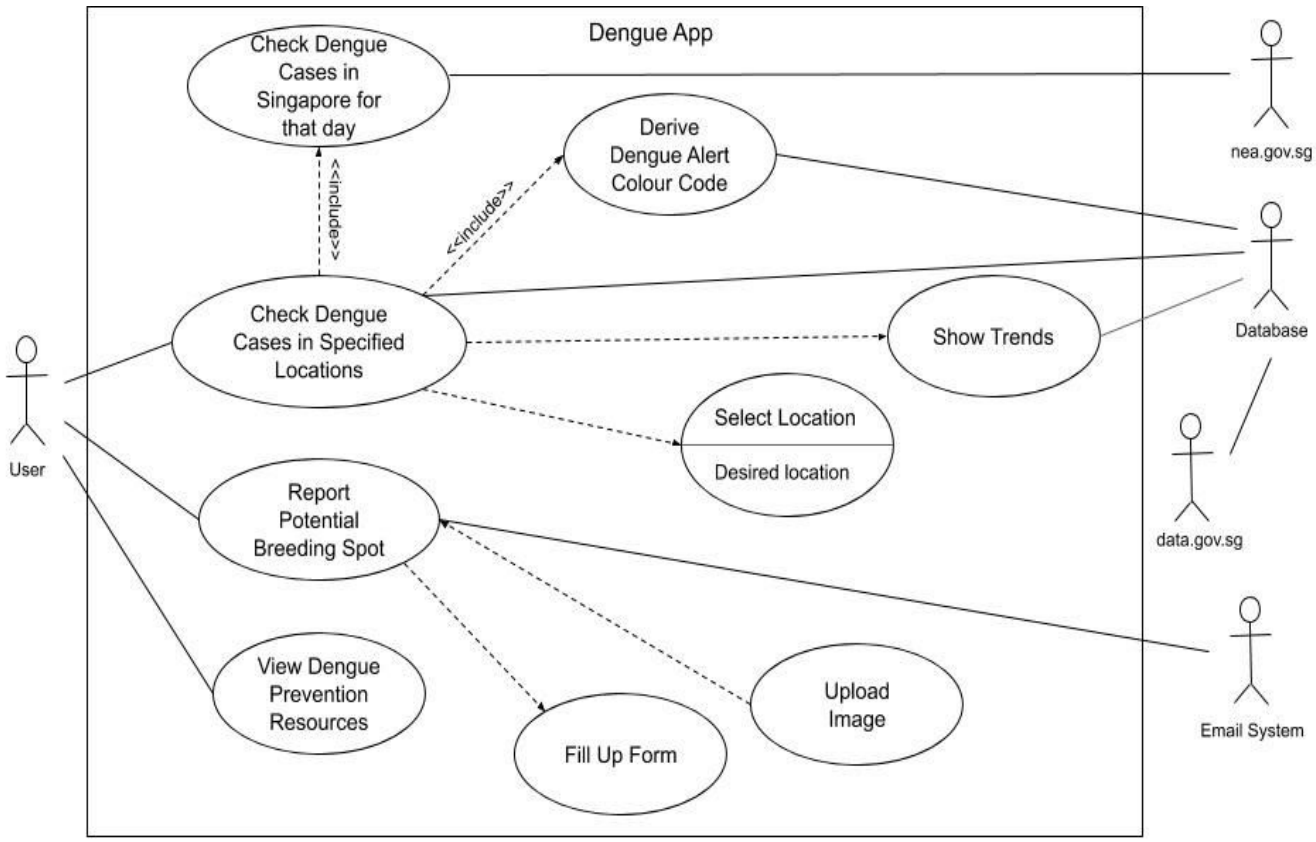
Flow of Events:	1. The user is presented with an option to input a specific location 2. The app queries database for the dengue alert colour code of the specified location 3. The app changes the colour of its interface to match the derived colour from the database
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	5.1		
Use Case Name:	View Dengue Prevention Resources		
Created By:	Wanyi	Last Updated By:	
Date Created:	08/09/20	Date Last Updated:	

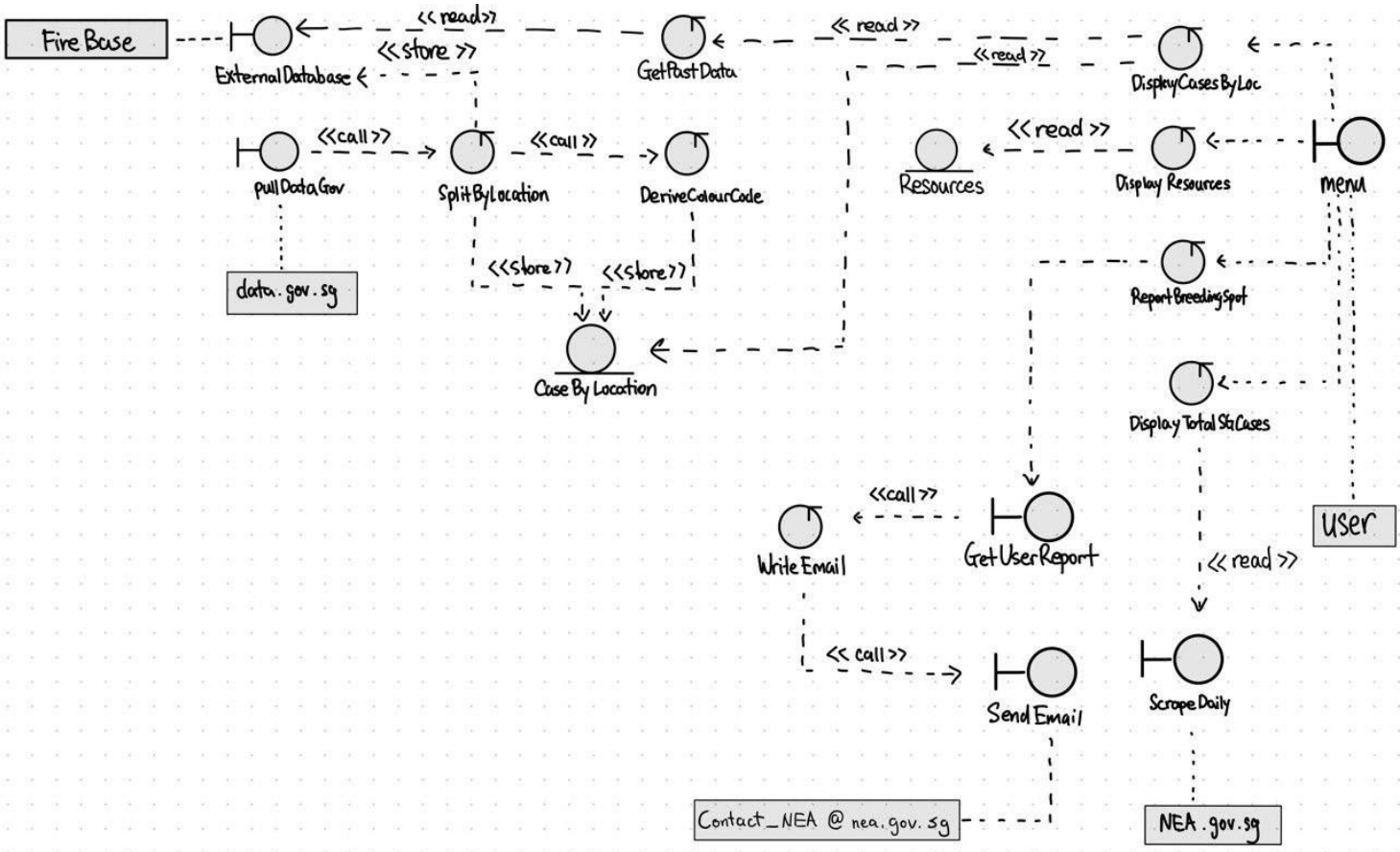
Actor:	The user is able to select from 4 different choices on the app interface to view different information regarding dengue prevention, recognizing dengue symptoms, and a hotline connecting them to NEA.
Description:	The app will display the information depending on the choice selected by the user. These information have been added into the app beforehand as text or graphics.
Preconditions:	1. The app boots successfully 2. The user has selected a choice of resource to view
Postconditions:	1. The interface displays text, graphics or flowcharts according to the choice made by the user
Priority:	Medium
Frequency of Use:	1-2 times per day

Flow of Events:	1. The user is presented with various options to select from 2. The app displays the information that we have input for each selection
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

5. USE CASE DIAGRAM

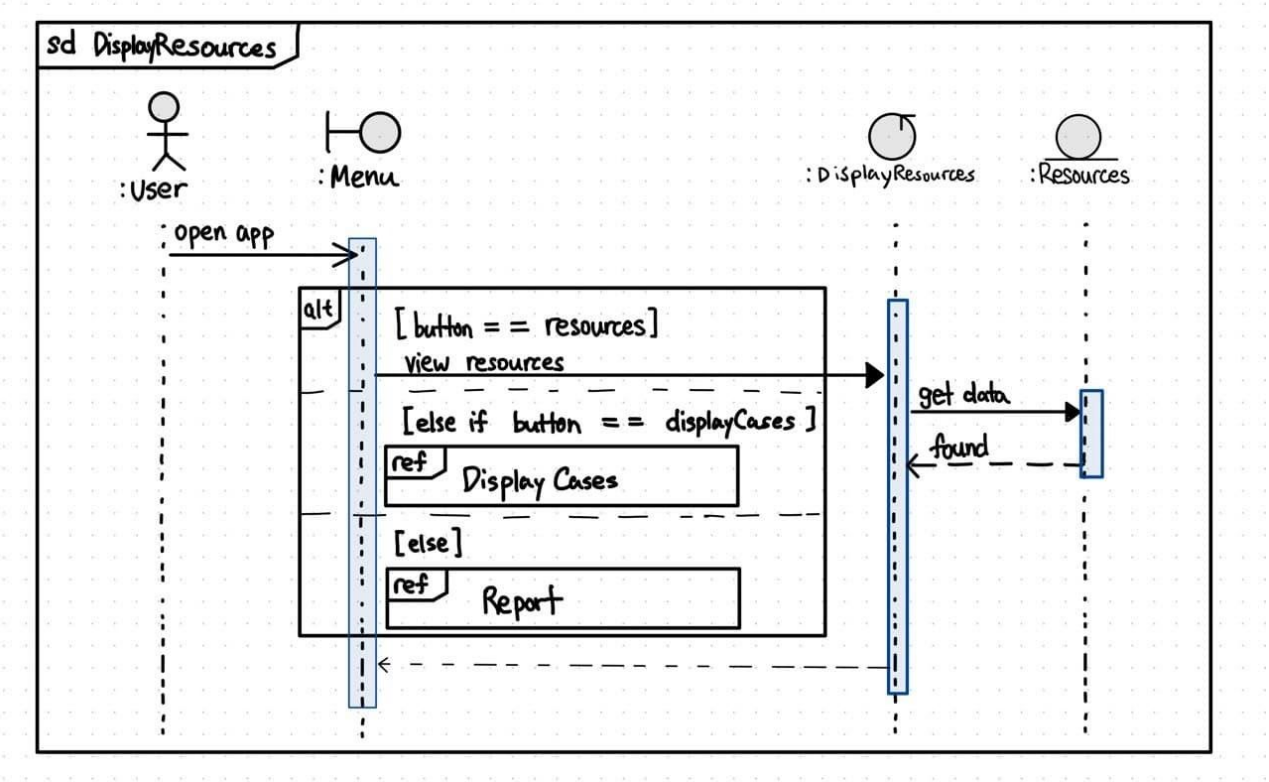


6. CLASS DIAGRAM

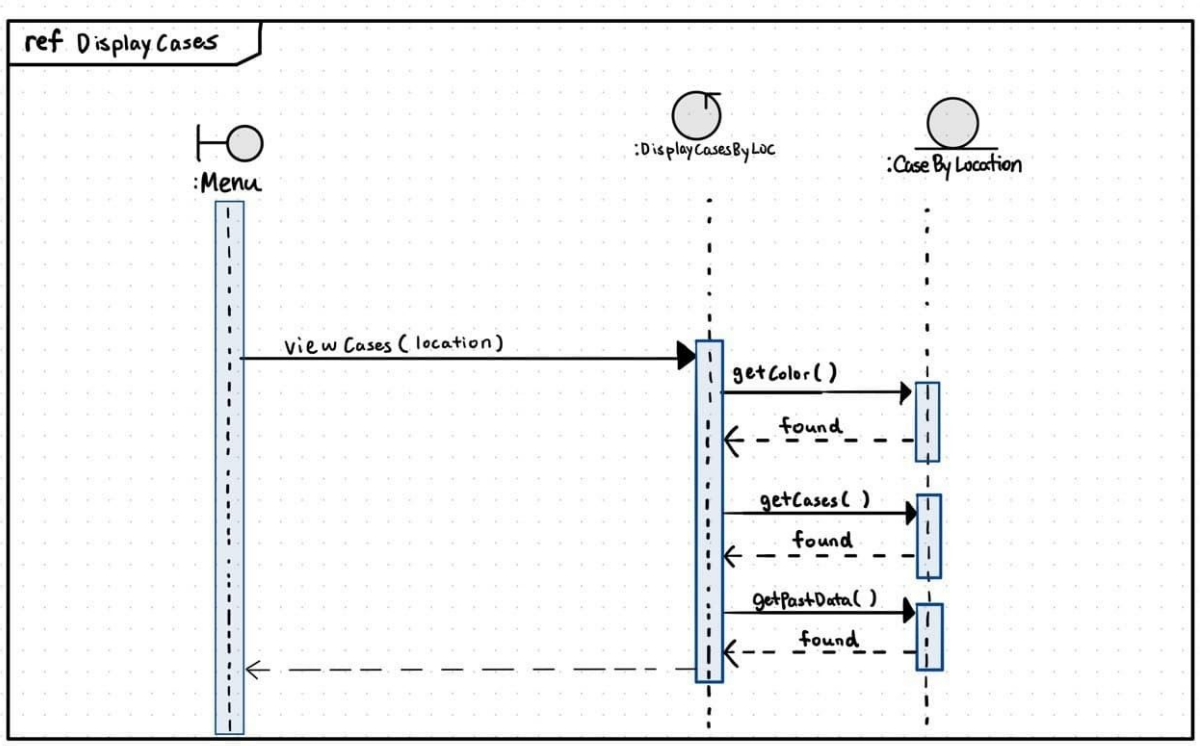


7. SEQUENCE DIAGRAMS

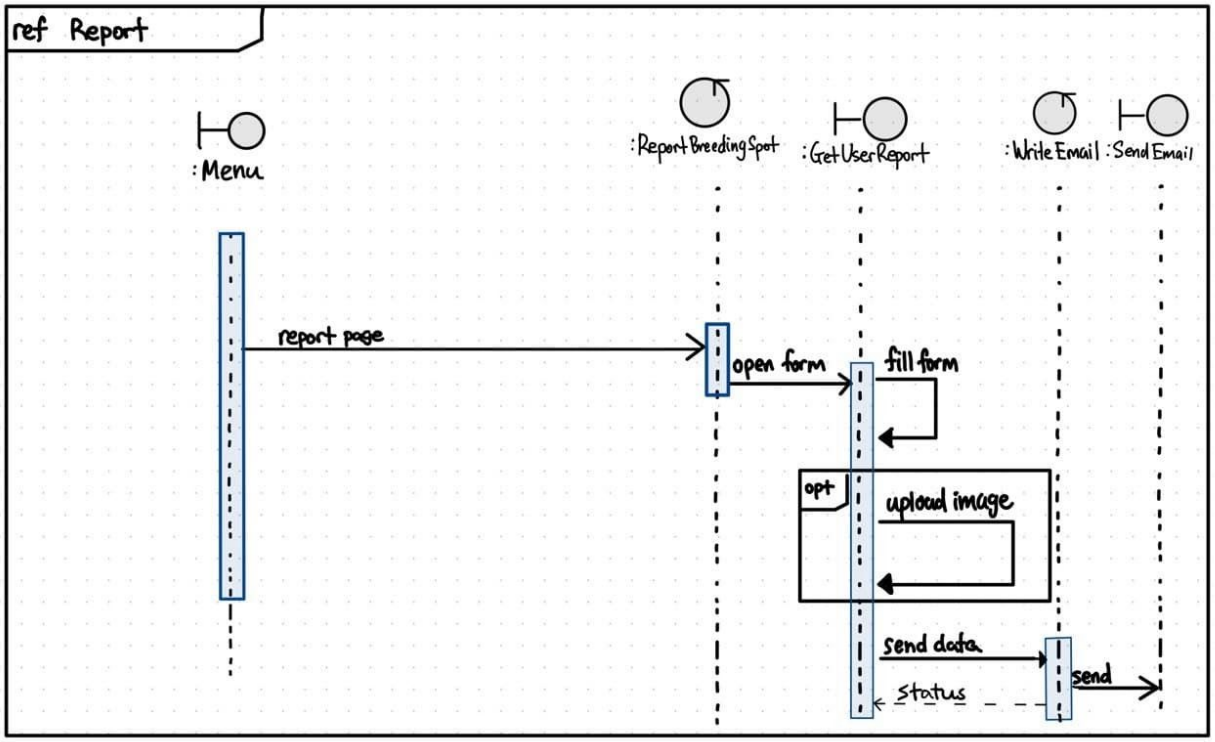
Sequence Diagram for Use Case ‘View Dengue Prevention Resources’



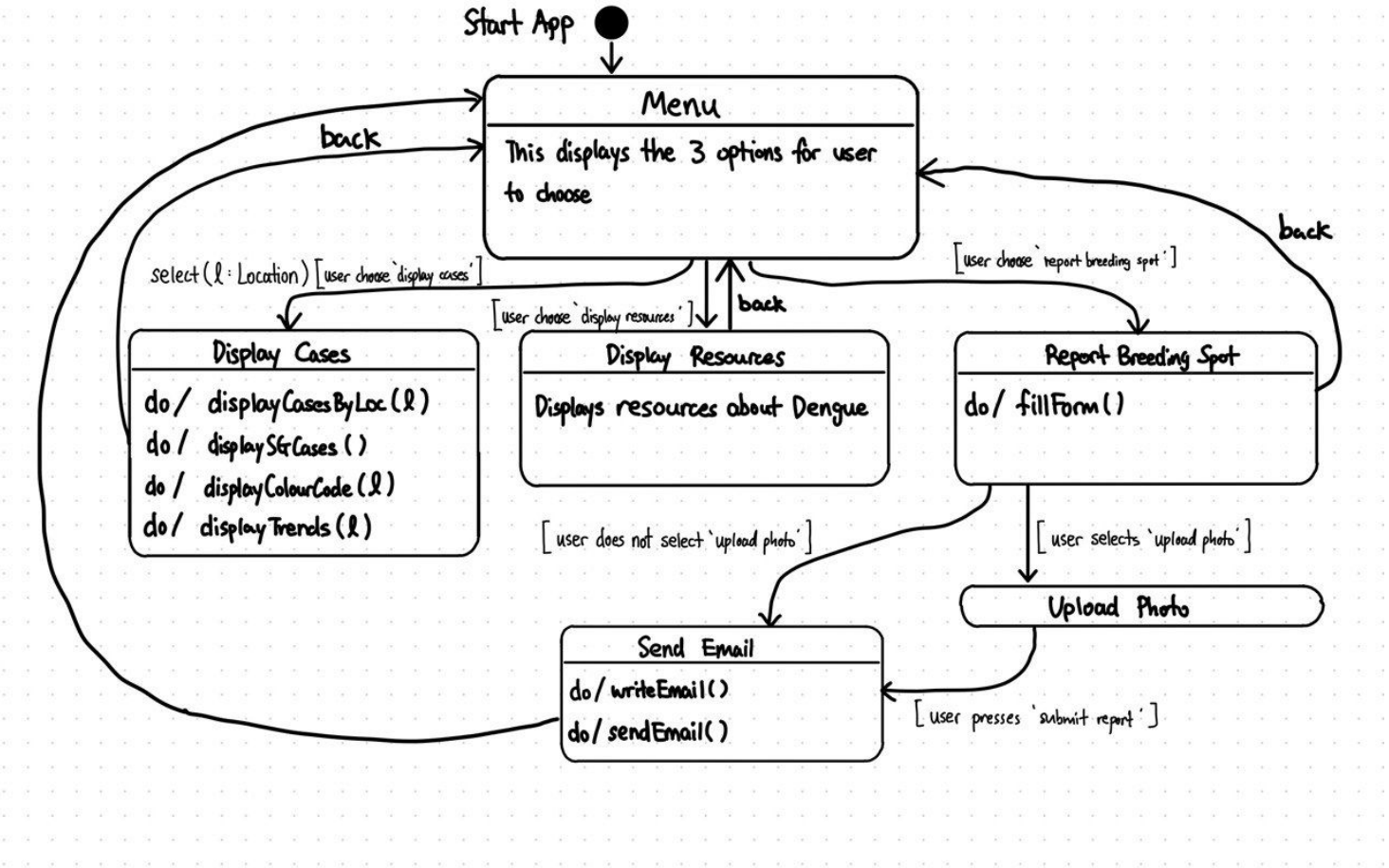
Sequence Diagram for Use Case ‘Check Dengue Cases in Specified Locations’



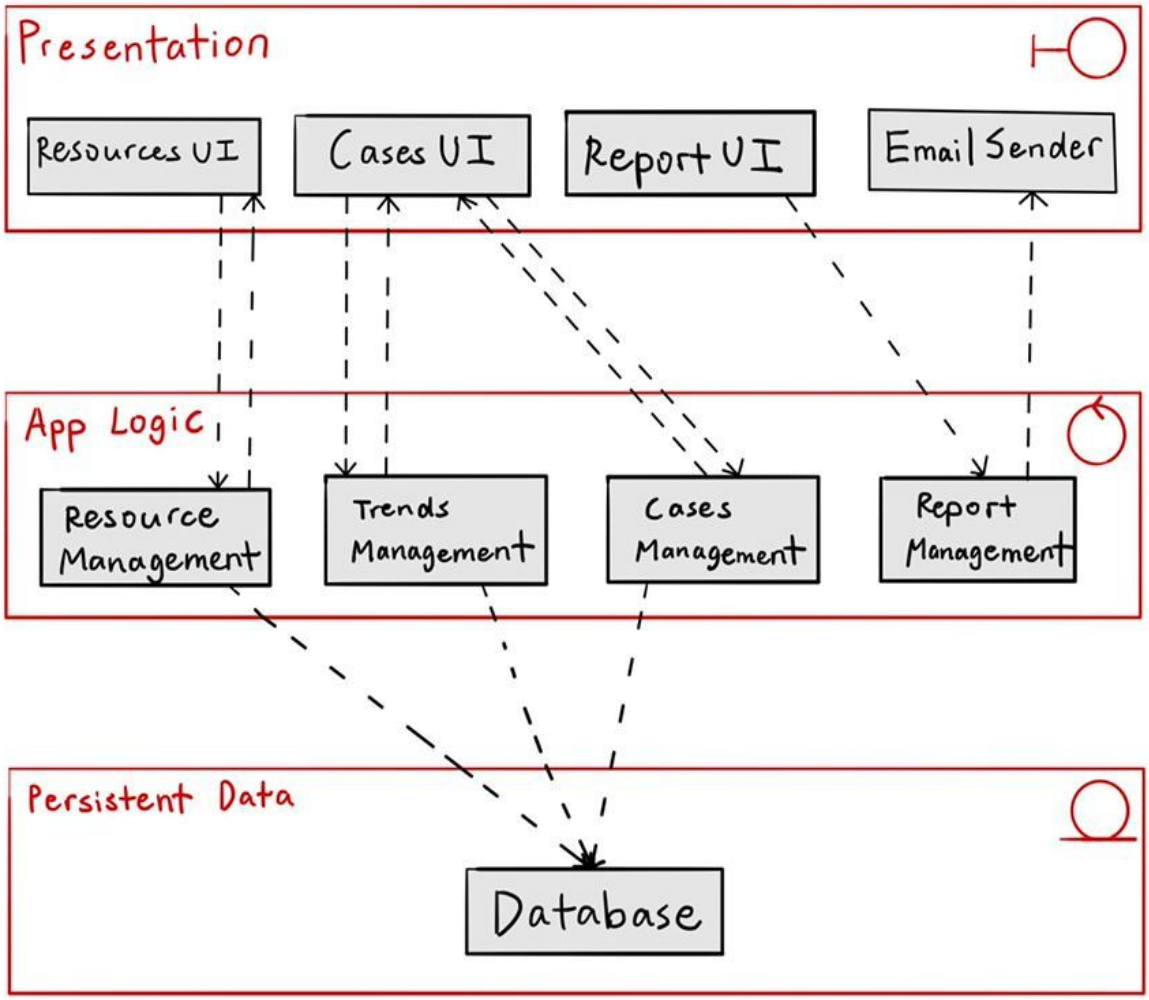
Sequence Diagram for Use Case 'Report Potential Breeding Spot'



8. STATE MACHINE DIAGRAM FOR SYSTEM USER INTERFACE



9. SYSTEM ARCHITECTURE



3-tier architecture diagram

10. GOOD SOFTWARE ENGINEERING PRACTICES

3-Tier Architecture Style:
We chose the 3-tier architecture style due to its many advantages, such as reusability of lower layer components while more upper layers are added, as well as the ease of introducing new functionalities and components.

Low Coupling and High Cohesion:
Similar functionalities are grouped together in our architectural diagram. This is to reduce communications between different components, hence promoting low coupling. Different components, each with a single responsibility, work together to achieve high cohesion. This also fulfils the single responsibility principle.

Separation of Concerns:
Different concepts are separated into different components in our architecture diagram.

Principle of Least Knowledge:
Each component in our architecture diagram has minimal knowledge of other components.

11. TEST CASES AND RESULTS

11.1 Location Case Number

Test Input	Expected Output	Actual Output	Result
Beach Rd (Blk 6)	To match database value	Matches database value	PASS

11.2 Singapore Case Number

Test Input	Expected Output	Actual Output	Result
(blank)	To match database value	Matches database value	PASS

11.3 Search Functionality

Test Input	Expected Output	Actual Output	Result
(blank)	No list items	No list items	PASS
a	A list full of addresses containing “a”	A list full of addresses containing “a”	PASS
Jln ampas	Only one item “Jln Ampas”	Only one item “Jln Ampas”	PASS
zzxxyy	No list items	No list items	PASS

11.4 Send Report as Email

Test Input	Expected Output	Actual Output	Result
Location: “ABC” Brief Description: “XYZ” One photo attached.	An email is drafted with “ABC”, “XYZ” as the email body. The photo is embedded after the email body.	An email is drafted with “ABC”, “XYZ” as the email body. The photo is embedded after the email body.	PASS
Location: “ABC” Brief Description: “XYZ” No photo attached.	An email is drafted with “ABC”, “XYZ” as the email body. There is no embedded photo after the	An email is drafted with “ABC”, “XYZ” as the email body. There is no embedded photo after the	PASS

	email body.	email body.	
Location: [Empty] Brief Description: [Empty] No photo attached.	An email is drafted with no location and no description in the email body. There is no photo embedded after the email body.	An email is drafted with no location and no description in the email body. There is no photo embedded after the email body.	PASS
Location: “Example1” Brief Description: [Empty] No photo attached.	An email is drafted with the location listed as “Example 1”. There is no description in the email body. There is no photo embedded after the email body.	An email is drafted with the location listed as “Example 1”. There is no description in the email body. There is no photo embedded after the email body.	PASS
Location: [Empty] Brief Description: “Example2” No photo attached.	An email is drafted with no location listed. There is a description “Example2 ”in the email body. There is no photo embedded after the email body.	An email is drafted with no location listed. There is a description “Example2 ”in the email body. There is no photo embedded after the email body.	PASS

12. USER INTERFACE MOCKUPS

Fig 1. Splash screen and Welcome screen

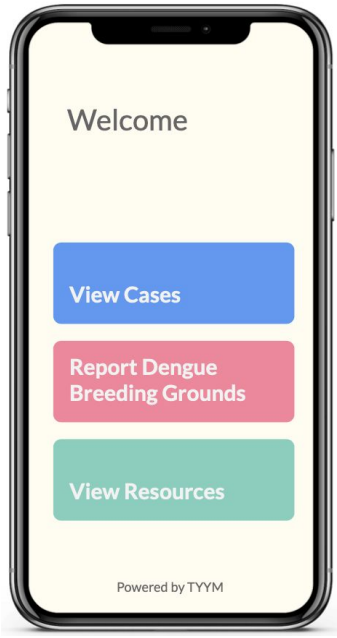


Fig 2. Main menu screen

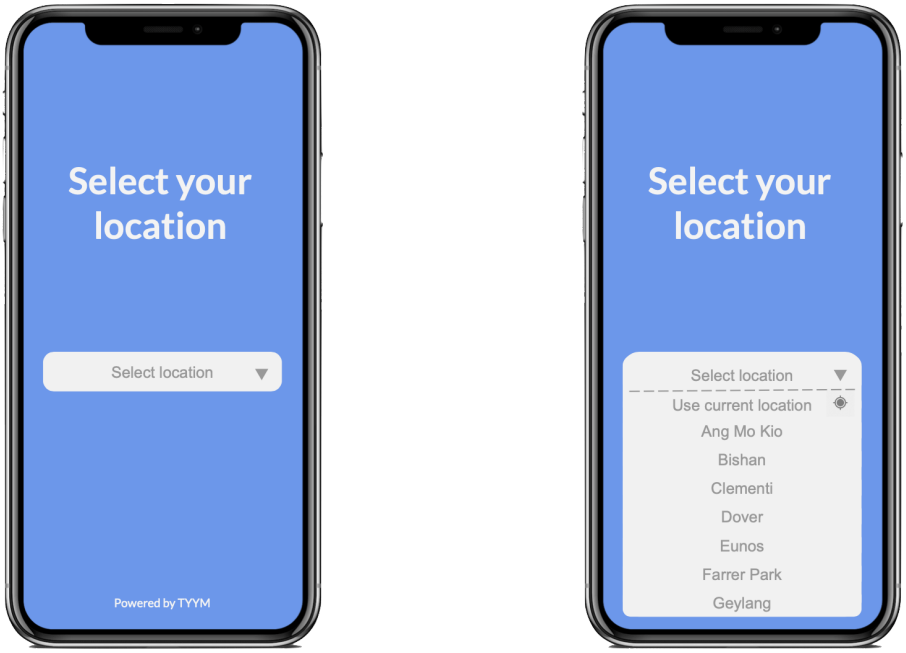


Fig 3. ‘Select location’ screen under the ‘View Cases’ function

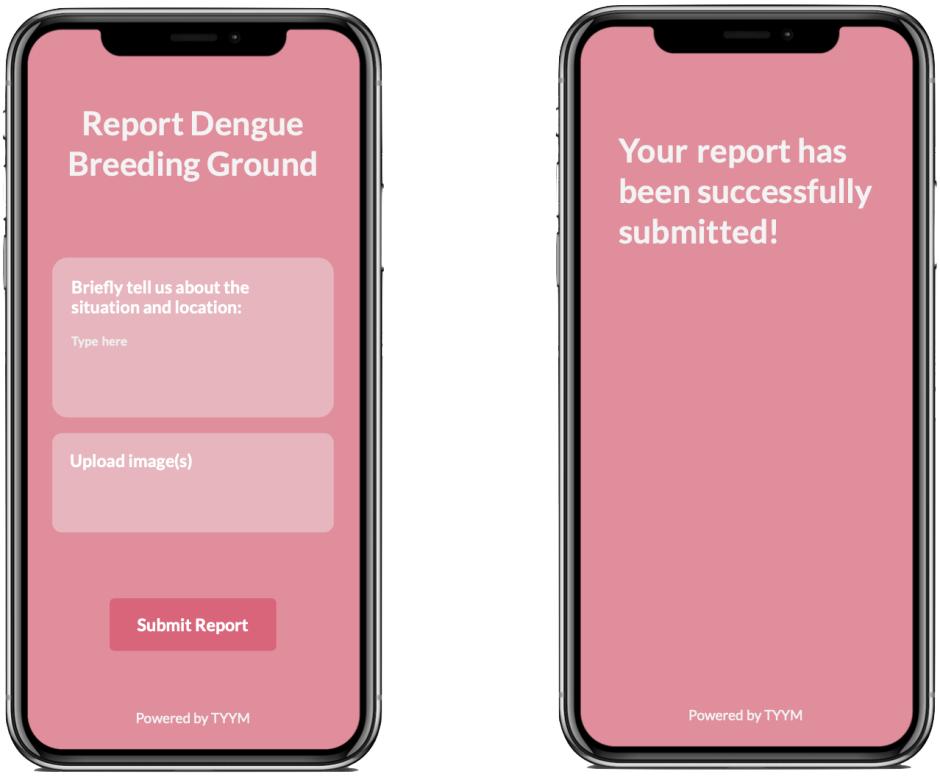


Fig 4. ‘Report Breeding Ground’ screen

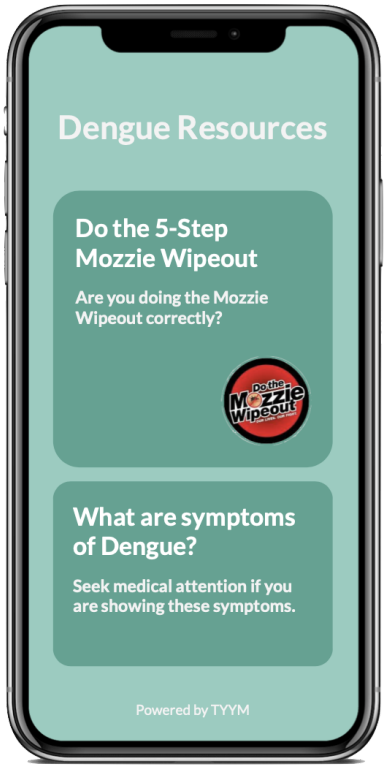


Fig 5. ‘View Resources’ screen

13. APP SCREENSHOTS

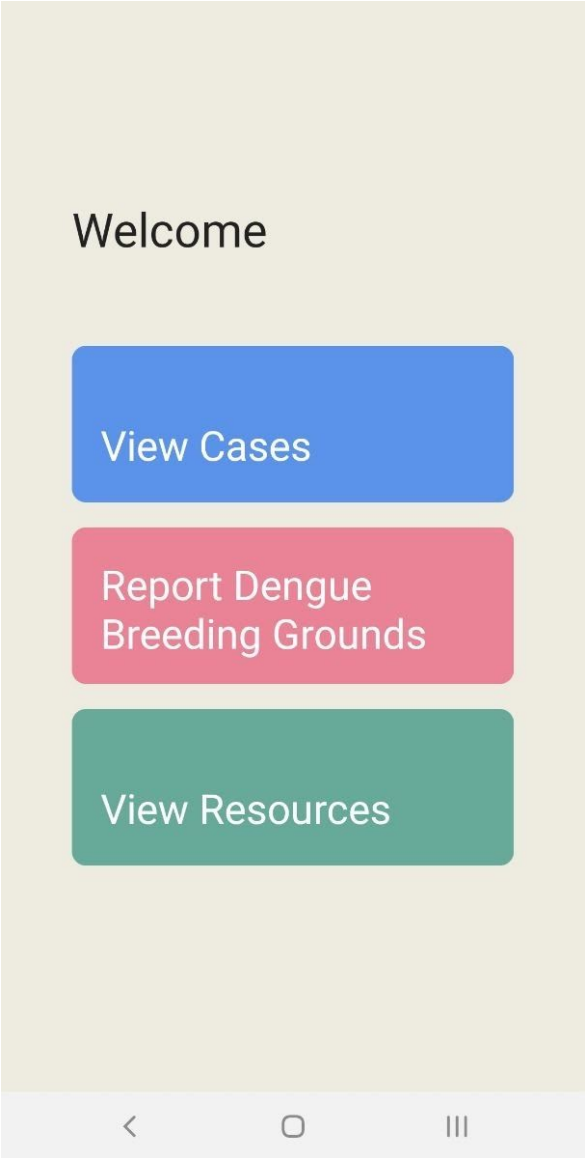


Fig 6. Home Screen

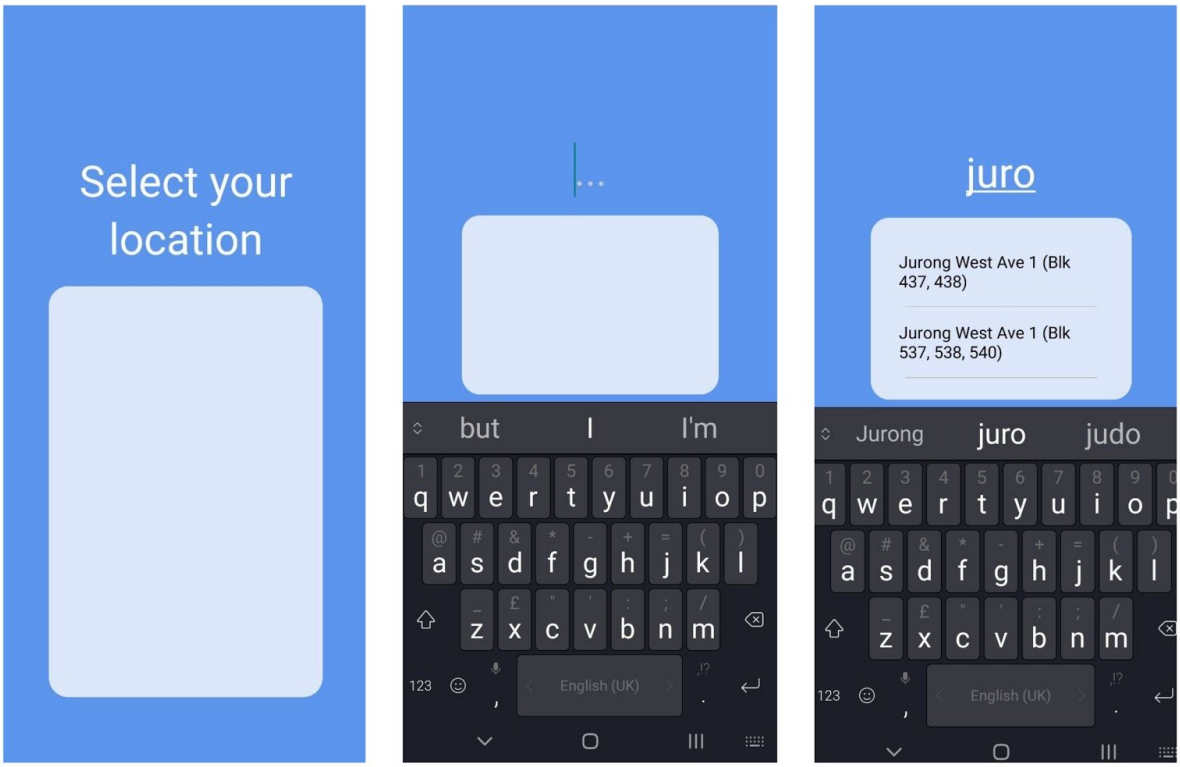


Fig 7. Search and ‘Select location’ Screen from ‘View Cases’

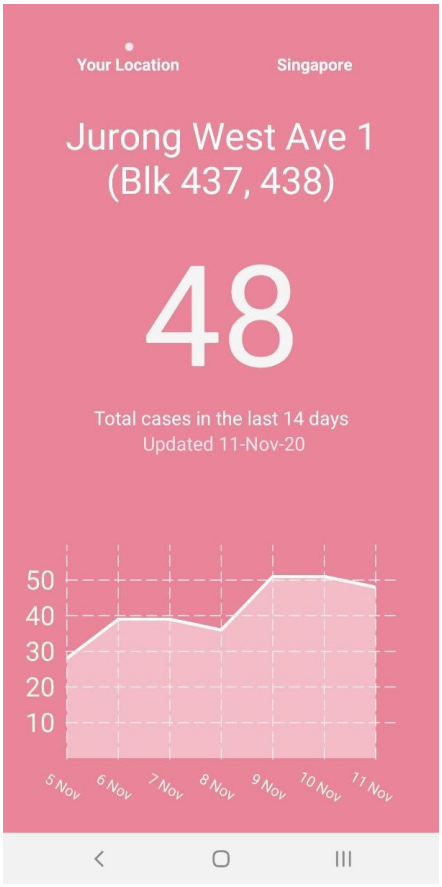


Fig 8. View Cases by location (Red Alert) Screen

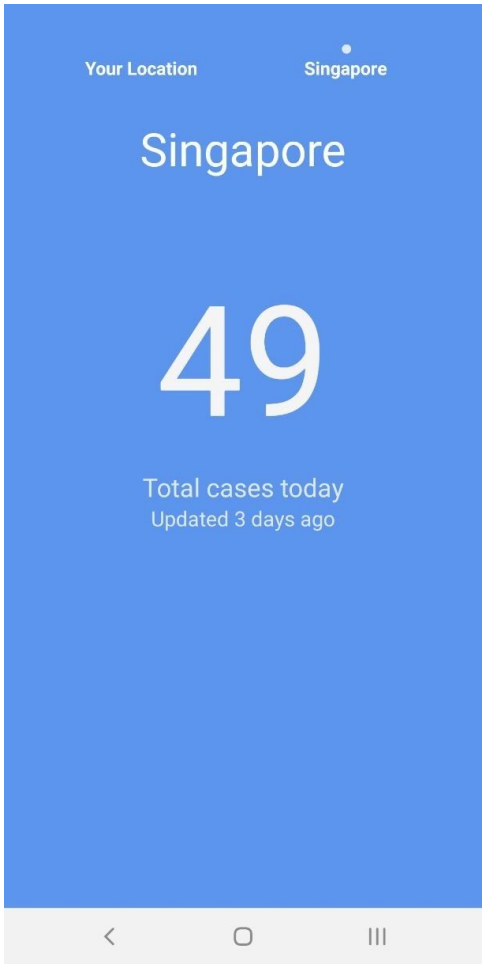


Fig 9. View total cases in Singapore Screen

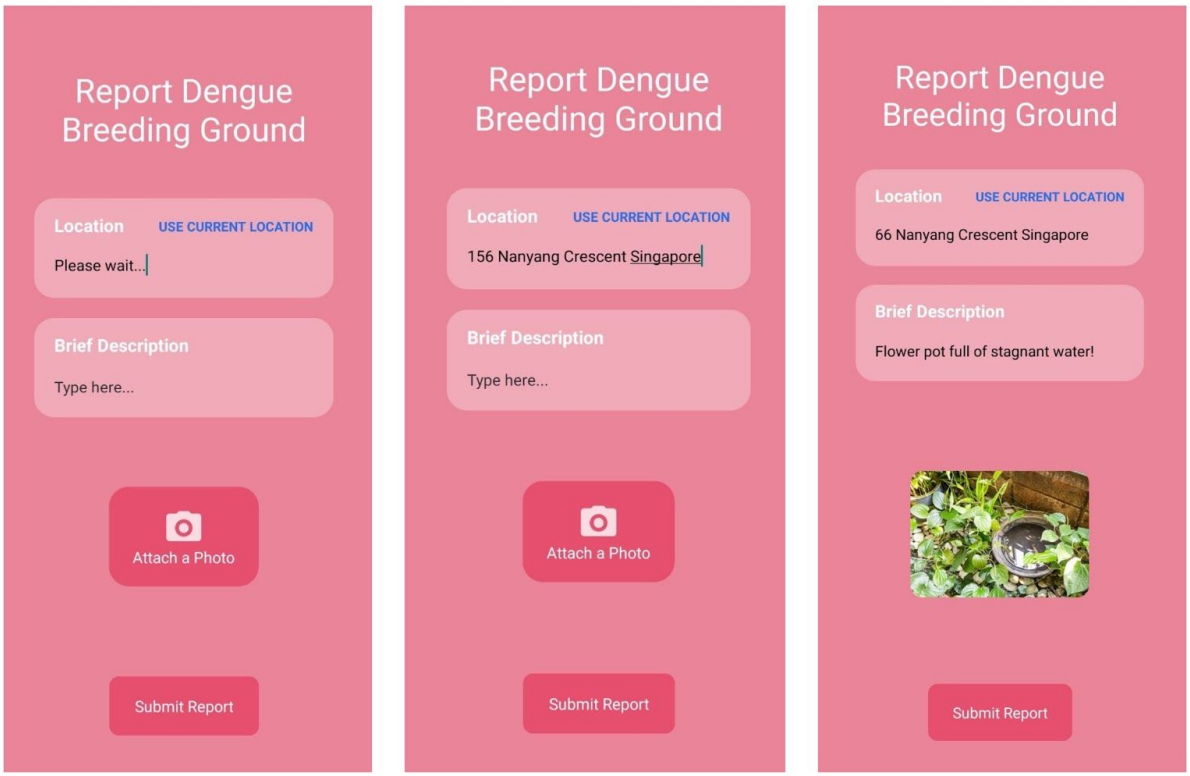


Fig 10. 'Report Dengue Breeding Ground' Screen

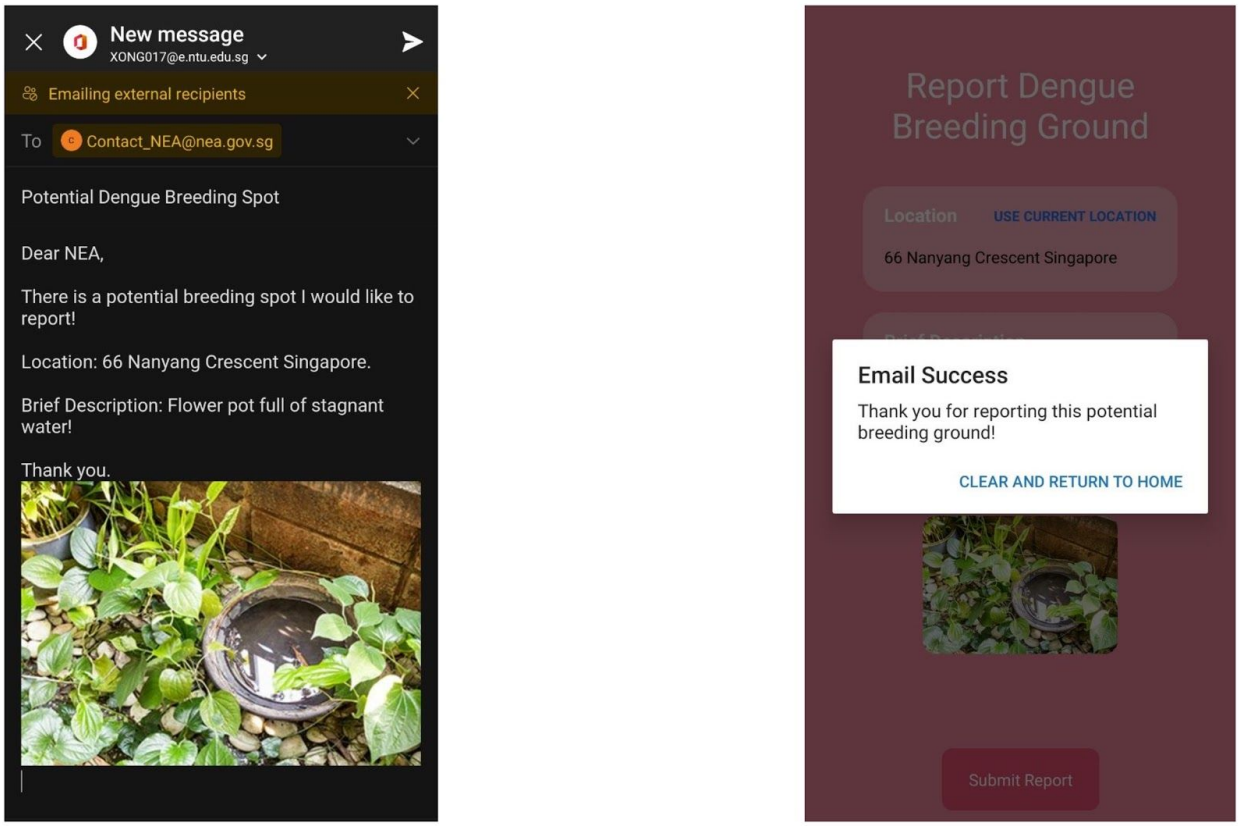


Fig 11. Email Sent to NEA from ‘Report Dengue Breeding Ground’

Fig 12. ‘View Dengue Resources’ Screen

APPENDIX A: MEETING MINUTES

Lab Review 1 Minutes:

Review of Past Agenda

-	**Application or functions of the app**	● Remove hotspot map idea to make room to make room for other applications that have more value ○ Suggested applications: ■ Tailoring news to the location ■ Sharing of app statistics with friends and family ■ Reporting / camera function
-	**Functional/Non functional Requirements**	● Change the reliability clause under non-functional requirements ● Add an additional non functional requirement to replace the reliability point ● Review the functional requirements after adding new functionalities to the app
-	**Use case Diagram**	● Reevaluate the use-case diagram according to the new functions
-	**Mock-up diagram**	● Change the mock-up diagrams (for the interface) to fit the new desired functions.
-	**Additional comments by Prof**	● Ensure that our project is not too simple in comparison to other groups ● Stick through with minimalist design all the way ● Reliability = (working time periods or units of system / 1 - time taken to repair) ● Hotspot map function could alternatively be a journey planner ● Use an external interactor (charting function under excel or android)

Next Agenda (0930-1230)

0930-1030	**Additional app functions and application ideas**	● Come up with more ideas for the app. These ideas should not be too complex and should be added on cautiously ● Decide on the time period of data to show and when to update
1030-1100	**Use-case diagram Mock up**	● Adjust based on input ideas
1100-1230	**Lab 2 Deliverables**	● Complete Use Case diagram ● Use Case descriptions ● Class diagram of entity classes ● Key boundary classes and control classes ● Sequence diagrams of some use cases ● Initial Dialog map

Lab Review 2 Minutes:

Review of Past Agenda

0945-1021	**Application ideas**	● Checking the daily cases of dengue ● View the 14 day trend of dengue based on location ● Reporting system with automated email to NEA, added camera functions ○ Whether location can be inputted/written into the form ● Optional: Hotspot map (color coded) that shows where dengue is more frequent in Singapore. Not interactable
0950-1030	**Functional Requirement**	● Adjusted clauses based on new ideas ● Ensure user-case tallies with Functional requirement
1030-1104	**Use-case Diagram**	● Darren email prof to ask regarding 'include'. Check dengue statistics -> select location ● Darren email prof to ask regarding connection between two actors ● Removed 'View Hot Spot map' ● Added 'derived dengue color code', connected to data.gov.sg ● Added 'include' to select location ● Added new actor 'database' that will be connected to show trends ● Added report potential breeding spot, extended by 'upload image', include 'fill up form' ● Added 'View dengue prevention resources'
1104-1120	**Lab 2 Deliverables**	● Complete Use Case diagram ● Use Case descriptions ● Class diagram of entity classes ● Key boundary classes and control classes ● Sequence diagrams of some use cases ● Initial Dialog Map
-	**Mock-up**	● Lin an working on it, to be done before next meeting
-	**Use case Description**	● To be done before next meeting

Next Agenda (Wed Sept 9 1330-1600)

1330-1400	**Mock-up, Use case description review**	
1400-1530	**Lab 2 Deliverables**	● Complete Use Case diagram ● Use Case descriptions ● Class diagram of entity classes ● Key boundary classes and control classes ● Sequence diagrams of some use cases ● Initial Dialog Map

Lab Review 3 Minutes:

Review of Past Agenda

1330-1400	**Review of Case description**	● To be updated based on future updates
1400-1445	**Entity class, conceptual model**	● Created multiple boundary classes, control classes and entity classes from the use case description and data dictionary ● 4 main control classes to carry out the functions of app ● Created extDatabase to pull from FireBase ● SplitbyLocation(street names) is a pre-processing step that will then channel the processed data into extDatabase
1445-1520	**Class Diagram of entity classes**	● CaseByLocation ○ -color: string ○ -addresses: string[...] ○ -cases: int[...] ○ -Past data int[7] ○ getColor(): String ○ setColor(String variable name): void ○ +getPastData ():int[7] ○ +setPastData(Int arr variable): void ○ +getCases(string address); int ○ +getCasesdict(map(string address, int case)):void ● Resources ○ hotlineHum:string ○ mozzieWipeoutSteps: String ○ medicalAdvice: String

-

Next Agenda (Wed Sept 9 1330-1600)

0930-1030	**Lab 2 Deliverables**	● Sequence diagrams of some use cases ● Initial Dialog Map

-

Lab Review 4 Minutes:

Review of Past Agenda

0930-1030	**Sequence diagram**	● Need to clarify how to use reference appropriately in sd ● Created a sequence diagram for each main function and reference the other functions in each sd ● Optional flow used for upload image
1030-1130	**Dialog map**	● Created dialog map with each state being a main function
-	**-**	
-	**Comments**	

Next Agenda (TBC)

-	**Review of Lab 2 Deliverables**	

-	**Lab 3 Deliverables**	● Complete Use Case model
		● Design Model
		  ○ Class diagram
		  ○ Sequence diagrams
		  ○ Dialog map
		● System architecture
		● Application skeleton

Lab Review 5 Minutes:

Review of Past Agenda

0930-1030	**Touch up on past deliverables**	● Changed layered to 3-tier architecture ● Begin testing app with compiled test cases ● Create app demo (by xx) with voiceover

1030-1130	**Lab 3 Deliverables**	● Complete Use Case model
		● Design Model
		  ○ Class diagram
		  ○ Sequence diagrams
		  ○ Dialog map
		● System architecture
		● Application skeleton

-

-

Next Agenda (TBC)

-

-