

# **Software Requirements Specification**

**for  
DengueGo!**

**Version 1.0 approved**

**Prepared by**

**Anil Ankitha, Chen Xueyao, Shyam Sundar, Kangxin Wang, Wong Ze Ming**

**Black and Yellow**

**14<sup>th</sup> April 2021**

<b>Introduction</b>	<b>5</b>
Purpose	5
Document Conventions	5
Intended Audience and Reading Suggestions	5
Product Scope	6
References	6
<b>Overall Description</b>	<b>7</b>
Product Perspective	7
Product Functions	7
User Classes and Characteristics	8
Operating Environment	8
Design and Implementation Constraints	8
User Documentation	9
Assumptions and Dependencies	9
<b>External Interface Requirements</b>	<b>10</b>
User Interfaces	10
UI Mockups	13
Hardware Interfaces	14
Software Interfaces	14
Communications Interfaces	14
<b>System Features</b>	<b>14</b>
User Registration	14
Description and Priority	14
Stimulus/Response Sequences	14
Functional Requirements	14
User Login	16
Description and Priority	16

Stimulus/Response Sequences	16
Functional Requirements	16
Forgot/Change Password	16
Description and Priority	16
Stimulus/Response Sequences	16
Functional Requirements	16
Map of Dengue Clusters	17
Description and Priority	17
Stimulus/Response Sequences	17
Functional Requirements	17
Dengue Risk Calculator	19
Description and Priority	19
Stimulus/Response Sequences	19
Functional Requirements	19
Risk-reducing Measures	20
Description and Priority	20
Stimulus/Response Sequences	20
Functional Requirements	20
Proximity Notification	21
Description and Priority	21
Stimulus/Response Sequences	21
Functional Requirements	21
<b>Other Non-Functional Requirements</b>	<b>21</b>
Performance Requirements	21
Security Requirements	22
Software Quality Attributes	22
Reliability	23
Usability	23
<b>Other Requirements</b>	<b>24</b>

Appendix A: Glossary	24
Data Dictionary	24
Appendix B: Analysis Models	26
Use Case Diagram	26
Use Case Descriptions	27
User Registration	27
User Login	28
Forgot Password	30
Dengue Risk Calculation	31
Proximity Notification	33
Safety Measures	34
Map of Clusters	36
Saving of location	37
View Infographics	38
Conceptual Class Diagram	40
Final Class Diagram	41
Sequence Diagrams	42
Login	42
Register	43
Forgot password	44
Home Screen	44
Information Screen	45
Survey Screen	46
Saved Location Screen	47
Notification Screen	47
Account Screen	48
Dialog Map	49
System Architecture and Design	50
Architecture	51

Design Patterns	51
Testing	53
Black Box Testing	53
Sign In	53
Verify OTP	54
Register account	57
Forgot Password	58
Change Password	59
Survey	60
Cluster Information	61
Saved Clusters	63
White Box Testing	64
Sign in	64
Verify OTP	66
Register new account	70
Forgot password	72
Change password	75
Dengue risk survey	77
Cluster information	79
Saved Clusters	81

# **1. Introduction**

## **1.1. Purpose**

Based on statistics that were collected between 2020 and 2021, we found out that dengue has caused more deaths than Covid-19 in Singapore.

Even amidst a global pandemic, dengue remains one of the main causes of death in Singapore, heralding its significance and relevance in our nation. Ergo, our group, Black and Yellow, have decided to tackle this issue and defend ourselves from our six-legged “friends”.

Our application, DengueGo! serves two main purposes. They are as follows:

1. The first allows users to find nearby clusters based on their mobile device’s GPS location. By using our app, users will be able to discover clusters and get notified if they are within proximity of one.
2. The second is to educate users on the ways they can protect their home from mosquitos by sending daily reminders.

## **1.2. Document Conventions**

Our document is structured in a way such that we have placed more emphasis on the above-mentioned two purposes that we hope to fulfill. However, we did not forgo other functions, such as the saving of user details, in our pursuit of fulfilment.

## **1.3. Intended Audience and Reading Suggestions**

This document is intended for developers, project managers, staff from the National Environmental Agency, users, and documentation writers. The rest of this SRS contains a description of the product, the product functions, the target user classes and their characteristics, the operating environment, the design and implementation constraints, user documentation, assumptions, the product’s various interfaces, system features, non-functional requirements as well as product requirements and attributes related to the product’s safety, security and software.

This document is best read in full in the order specified above for developers, project managers, NEA staff and documentation writers due to their specialised jargon.

For users, please refer to the relevant sections 1.4 Product Scope, 2.2 Product Functions, 2.4 Operating Environment and 3.1-4.2 which covers the product's interfaces and system features. Please refer to 5.1-5.5 for insight into the base requirements of this product in terms of performance, safety, security and software.

## **1.4. Product Scope**

DengueGo! aims to educate the public on potentially harmful practices that may subject them to dengue, thus we have incorporated a survey and notification service that may help to alleviate some of the factors that lead to mosquito breeding in homes. We have also implemented a cluster information system as a form deterrence against dengue. Entering the specified radius of a known dengue cluster will alert the user to the potential risks involved.

## **1.5. References**

Khalik, S. (2020, November 19). *More people have died from dengue than from Covid-19 in Singapore this year*. Straits Times.

<https://www.straitstimes.com/singapore/health/more-people-have-died-from-dengue-than-from-covid-19-in-singapore-this-year>

## **2. Overall Description**

### **2.1. Product Perspective**

DengueGo is an innovative mobile application that aims to protect Singaporeans from the titular disease that has plagued our nation for decades, dengue. Its main purposes are to deter the general populace away from known dengue clusters as well as provide daily reminders to prevent dengue from occurring in their homes. In order to fulfill its raison d'etre, we have utilised web scraping methods to extract data of Singapore's dengue clusters that NEA has benevolently provided on data.gov.sg. With this precious information, and our technical prowess, DengueGo is capable of displaying the full geographical map of Singapore, complete with dengue clusters ranging from high to low risk, depending on the number of cases. From here, the user is able to view and save cluster locations. Courtesy of NEA's data, our application will also notify and warn users when they travel within 3km of a known cluster, thereby safeguarding them harm.

### **2.2. Product Functions**

- User's current location may be tracked using GPS
- Product must show user on an interactive map in real-time, the approximate location of dengue clusters in Singapore
- Product must provide users with notification alerts when their current location is within 3 km of a known dengue cluster
- Product must provide users with information on how to reduce the risk of contracting Dengue with a daily checklist of steps to follow.
- Product must provide users with a calculated Dengue risk score based on user input and provide recommendations based on risk score.
- Product must provide specific users with a login function.



### **2.3. User Classes and Characteristics**

The 3 main user classes of this product are staff from the National Environmental Agency (NEA), persons with permanent residence in Singapore and hotel staff, all likely frequent users of this product. Tourists and work permit holders are smaller user classes of this product and may not use this product as often.

We anticipate that staff from the NEA will be sending out dengue cluster alerts and conducting spot checks on various neighbourhoods with the help of this application. The former function is a requirement specific to NEA staff.

Persons with permanent residence in Singapore are generally Singaporeans or PRs staying in HDBs, condominiums or landed property. We predict that in this user class, the majority of users will be middle-aged. Studies have shown that the highest incidence rate of dengue infection occurs in adults in the 25-44 years age group, showing that this age group is the most heavily impacted by dengue. In addition, users in this age group are more likely to be homeowners responsible for keeping their homes at low risk of breeding *Aedes* mosquitoes.

Similarly, we anticipate hotel staff frequently using the risk calculating function of this product to ensure that their premises are safe and provide customers with peace of mind. We also predict that they will be using the notification function frequently so that customers can take the necessary precautions when going to any area with potential Dengue clusters.

The accommodation for work permit holders is generally provided for by their employers. As a result, it is unlikely that they will make use of the risk calculating function of the product as it will be the responsibility of their employers. However, we anticipate frequent usage of the notification system.

Tourists are users who do not have permanent residence in Singapore and will likely only be staying in Singapore for a period of time. Thus, it is unlikely that they will make use of the risk calculating function of the product. The notification system will likely only be used during the period of their stay in Singapore. Therefore, tourists' usage of the product will not be significant.

### **2.4. Operating Environment**

Our software operates best on Android phones.

### **2.5. Design and Implementation Constraints**

Since the NEA website collects data in English, the application is only available in English.

Our application requires Internet access and GPS turned on to work.

## **2.6. User Documentation**

The mobile application is easy-to-use and intuitive as it emulates most mobile applications hence no user manual is given.

## **2.7. Assumptions and Dependencies**

The reliability and accuracy of the dengue clusters is based on the data obtained from the website

<https://www.nea.gov.sg/dengue-zika/dengue/dengue-clusters> and

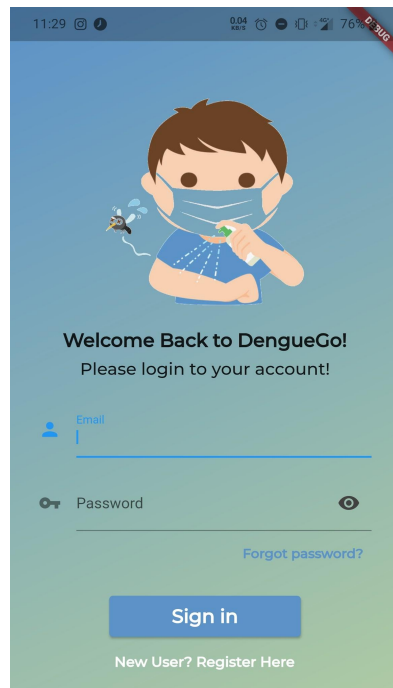
<https://www.nea.gov.sg/dengue-zika/dengue/dengue-clusters-under-surveillance>.

The google maps API has been utilized to view the clusters within the mobile application.

### 3. External Interface Requirements

#### 3.1. User Interfaces

Login Screen:



11:29 0.04 KB/s 76% 0.3UG

**Welcome Back to DengueGo!**  
Please login to your account!

Email

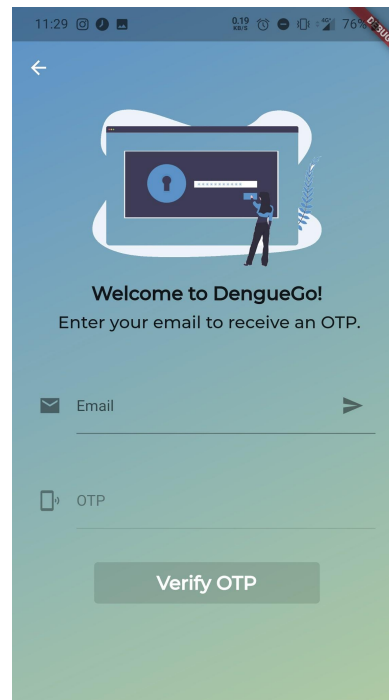
Password

[Forgot password?](#)

**Sign in**

New User? Register Here

Email Verification Screen:



11:29 0.19 KB/s 76% 0.3UG

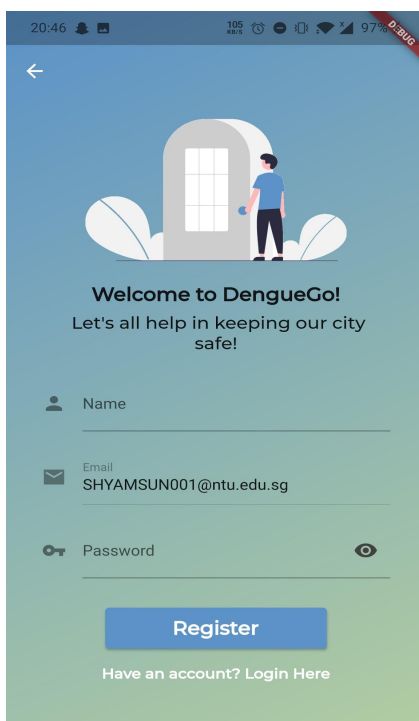
**Welcome to DengueGo!**  
Enter your email to receive an OTP.

Email

OTP

**Verify OTP**

Signup Screen



20:46 105 KB/s 97% 0.3UG

**Welcome to DengueGo!**  
Let's all help in keeping our city safe!

Name

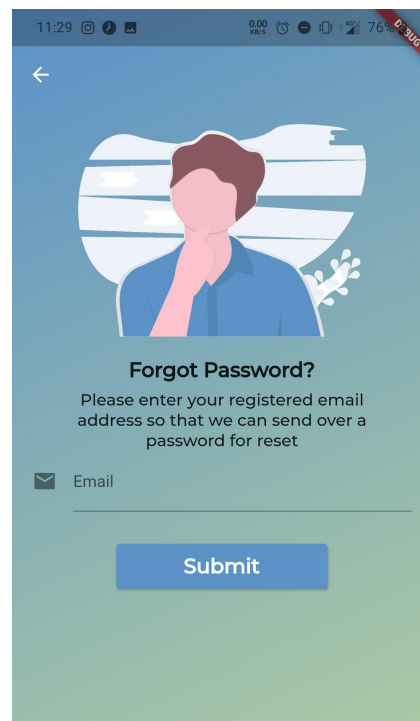
Email

Password

**Register**

Have an account? Login Here

Forgot Password Screen:



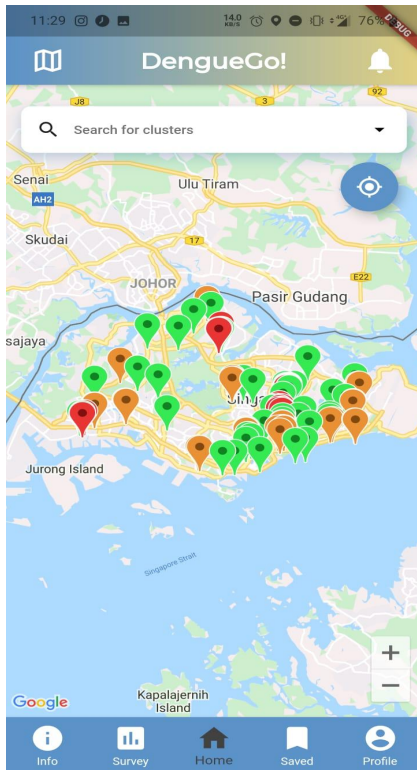
11:29 0.00 KB/s 76% 0.3UG

**Forgot Password?**  
Please enter your registered email address so that we can send over a password for reset

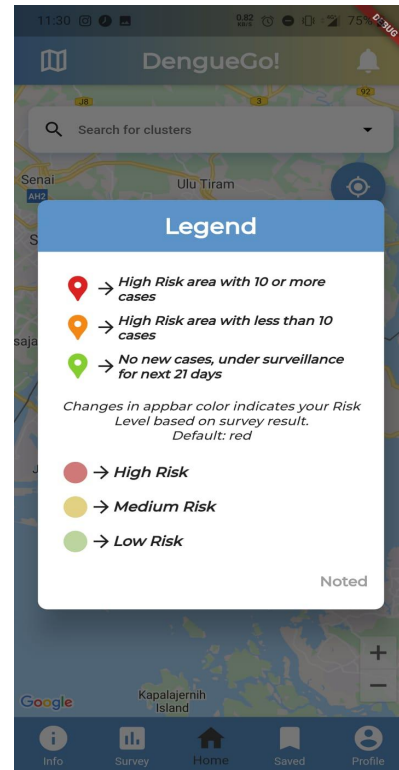
Email

**Submit**

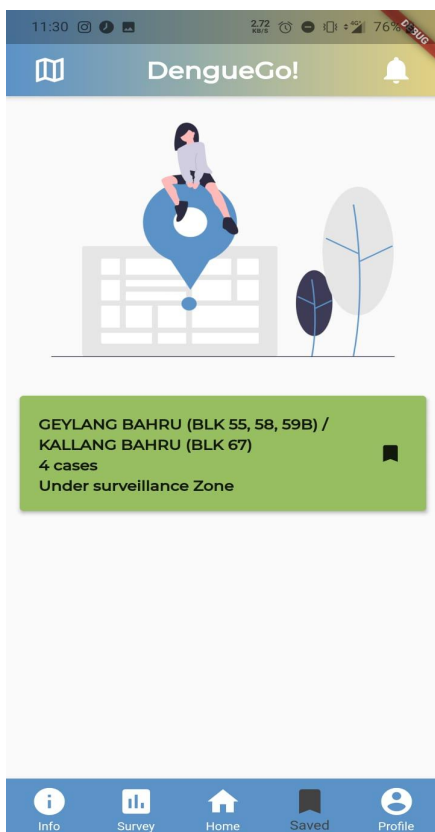
Main Screen:



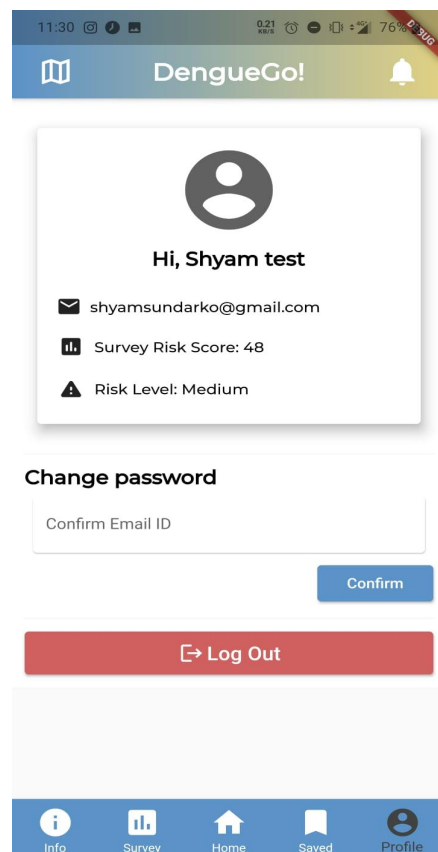
Legend Screen:



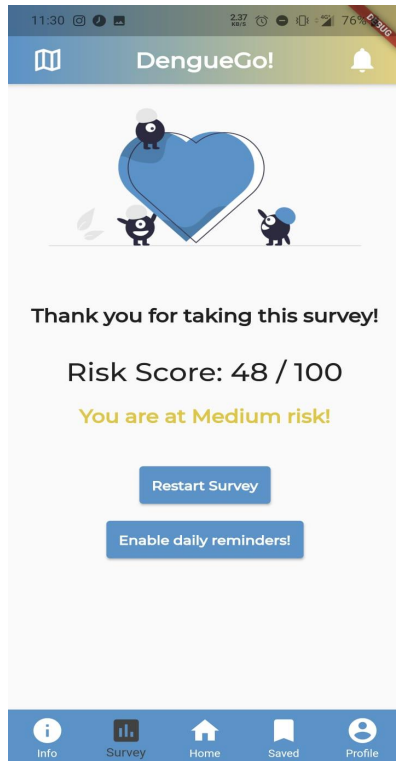
Saved Screen:



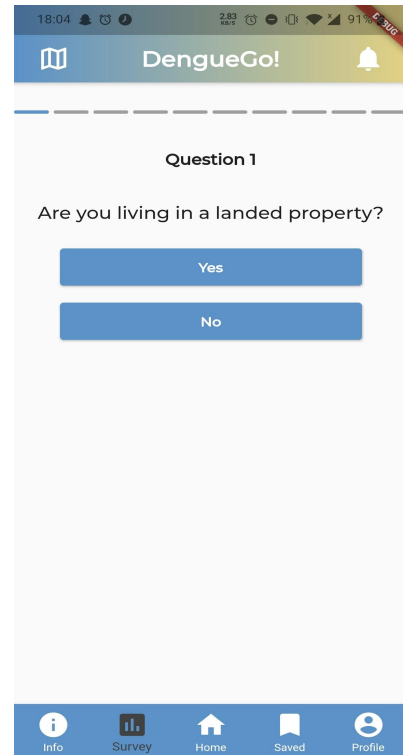
Account Screen:



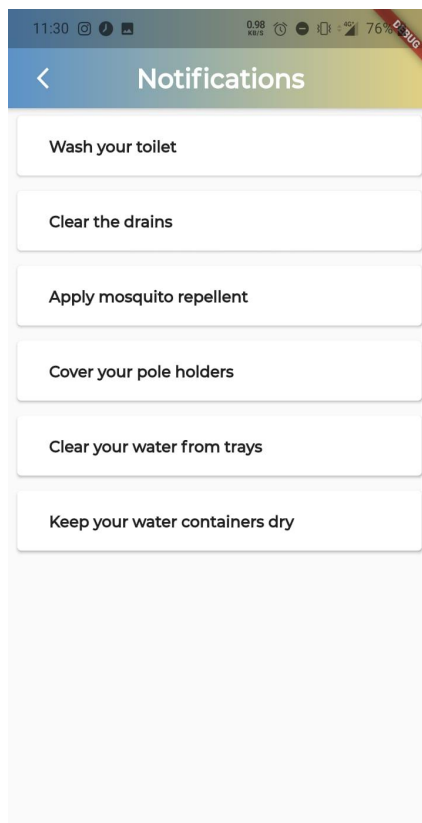
## Survey Result Screen



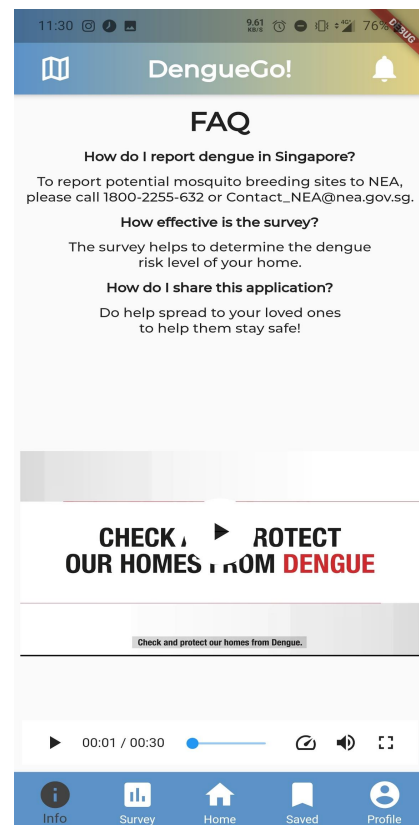
## Survey Question Screen



## Notification screen:



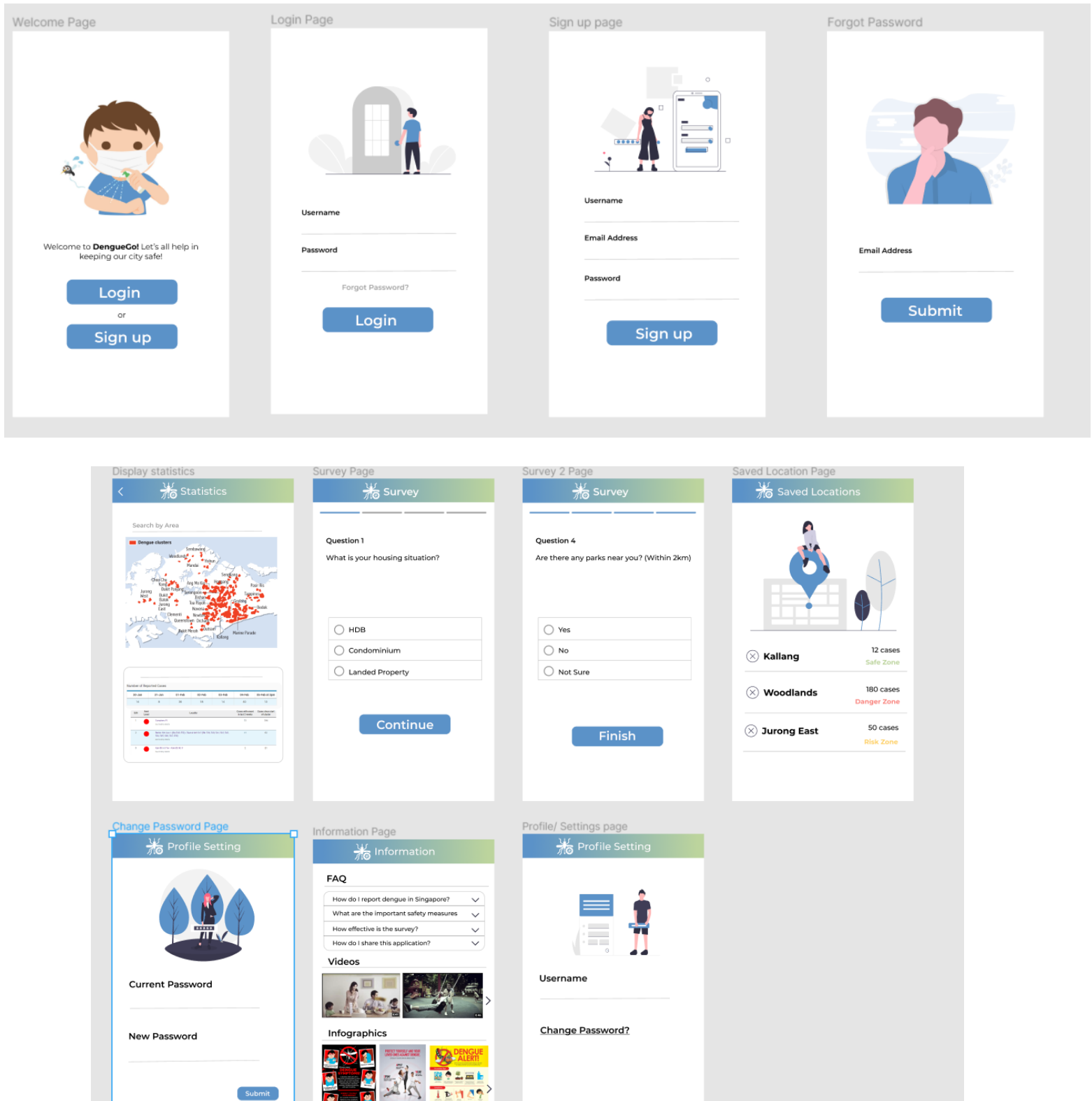
## Info Screen:



## 3.2. UI Mockups

Alternatively, here is the link to access our UI Mockups:

<https://www.figma.com/file/DuIp8z1fkmzYe3L7xfQLJr/DengueGo?node-id=141%3A0>



### **3.3. Hardware Interfaces**

DengueGo is functional on devices that have location services enabled to locate the user's current location.

### **3.4. Software Interfaces**

DengueGo is best used on Android devices. DengueGo uses Firebase for database and retrieves reminders based on the user's dengue risk level.

### **3.5. Communications Interfaces**

When users forget their password or when new users register for the application, users would need to access their email to reset and change their password.

## **4. System Features**

### **4.1. User Registration**

#### **4.1.1. Description and Priority**

This feature allows users to register for an account. This feature is of high priority.

#### **4.1.2. Stimulus/Response Sequences**

Users can register with a valid email and press send for an OTP to be sent to their email address. Then, they can key in the OTP they received and they would be brought to the sign-up page where they would be prompted to enter their name and password. Once the registration is completed, users will be logged in and redirected to the app's Home Screen.

#### **4.1.3. Functional Requirements**

1. The user must be able to fill in their email.
2. The system must validate if the user has provided an email.
3. The system must validate if the user input for email is of the correct format.
4. The system must validate that the email does not currently exist in the database.

5. The system must send an OTP to the entered email.
6. The system must check whether the OTP the user provides is correct.
7. The system must redirect the user to the next screen where the user provides their name and password.
8. The system must validate that both 'name' and 'password' fields are filled in.
9. If there is an empty field, the system must display an error message.
10. The system must validate that the user input for "password" is of sufficient length.
11. If the password provided is not long enough, the system must display an error message.
12. The system must redirect users to the Home Screen after account registration is successful.



## **4.2. User Login**

### **4.2.1. Description and Priority**

This feature allows specified users to login to the application. This feature is of high priority.

### **4.2.2. Stimulus/Response Sequences**

Users can login with their specific email and password.

### **4.2.3. Functional Requirements**

1. Users must be able to log in with their email and password.
2. The system must validate that both fields are filled in.
3. The system must validate that both email and password fields are of correct format.
4. If the email is not of the correct format, the system must display an error message.
5. If the password is invalid, the system must display an error message.
6. The system must validate the email exists in the database.
7. If the password and email do not match, the system must display an error message.

## **4.3. Forgot/Change Password**

### **4.3.1. Description and Priority**

This feature allows existing users to reset their password through their emails. This function has medium priority.

### **4.3.2. Stimulus/Response Sequences**

Users can provide their email address to the system and reset their password through a link sent to their email.

### **4.3.3. Functional Requirements**

1. The system must allow users to enter their email.
2. The system must validate the field is filled in

3. The system must validate the email format is correct.
4. If the email format is incorrect, the system must display an error message.
5. The system must validate the email exists in the database.
6. If the email does not exist in the database, the system must display an error message.
7. The system must send users a verification email when a password change is requested.
8. System must redirect users to the main menu.
9. The system must allow users to reset their password through the sent link.
10. The system must display an error message if the user leaves the password field blank.
11. The system must display an error message if the user input for the password field does not satisfy requirements.

## **4.4. Map of Dengue Clusters**

### **4.4.1. Description and Priority**

This feature shows the real-time information of dengue clusters in Singapore marked out as color-coded dots on an interactive map, to be updated daily at 1:00 am. This feature also enables the user to search for cluster locations and view information regarding current dengue clusters. The user can also track their live location. Additionally, the user will be able to save the cluster so that its information can be viewed later. This feature is of high priority as one of the best ways to lower the risk of contracting dengue is to avoid places where there are known to be disease clusters, which will require information provided by this feature.

### **4.4.2. Stimulus/Response Sequences**

Users will first provide the application with GPS tracking permissions. Users will then navigate to the Home Screen, which, by default, displays an interactive map of Singapore with markers to indicate where dengue clusters are. Users can search for clusters through the dropdown box or by tapping on a marker. Users can save a cluster to their Saved Cluster List. Users can track their current GPS location. If there are known dengue clusters within 3 km of the user's current location, the system will send a notification.

### **4.4.3. Functional Requirements**

1. The system must be able to display current dengue clusters on an interactive map.

2. The dengue clusters must be displayed as:
  - 2.1.1.1. Red markers if the cluster location is of High Risk zone with under 10 cases
  - 2.1.1.2. Orange markers if the cluster location is of High Risk zone with less than 10 cases
  - 2.1.1.3. Green markers if the cluster location is under surveillance zone.
3. The user must be able to view the full map of Singapore with markers showing dengue cluster locations.
4. The user must be able to interact with the map.
  - 4.1. The user must be able to zoom in and out on the map.
  - 4.2. The user must be able to switch to the map's default view by tapping on the map.
  - 4.3. The user must be able to drag the map.
5. The system must be able to click on a button and access the user's current location on the map
6. The user must be able to select a cluster from a dropdown list of clusters.
7. The user must be able to select a cluster by tapping on the associated marker.
  - 7.1. The system must zoom into a cluster when it is selected.
8. The system must be able to display a cluster's information when a cluster is selected.
  - 8.1. The system must display a cluster's information when its associated marker is tapped.
  - 8.2. The system must display a cluster's information when it is selected through the drop down list.
  - 8.3. The system must display the following information of the selected marker:
    - 8.3.1.1. Name of cluster location
    - 8.3.1.2. Number of location cases
    - 8.3.1.3. Name of cluster
    - 8.3.1.4. Number of cluster cases

8.3.1.5. Name of zone the cluster belongs

8.3.2. The system must display the following information if the zone is under surveillance

8.3.2.1. Name of cluster

8.3.2.2. Number of cases

8.3.2.3. Name of zone the cluster belongs

## **4.5. Dengue Risk Calculator**

### **4.5.1. Description and Priority**

This feature is a questionnaire that users fill up to assess their risk of contracting Dengue in their place of residence. This feature has high priority.

### **4.5.2. Stimulus/Response Sequences**

The user will fill up a survey and the system will return a score out of 100 along with their approximate risk of contracting Dengue. It will also provide a list of reminders that are catered to the level of risk they are at.

### **4.5.3. Functional Requirements**

1. Users must be able to fill in the survey.
  - 1.1. The survey must contain only the following types of questions:
    - 1.1.1. Yes/No Questions
    - 1.1.2. Questions with scoring ranging from 0 to 10
    - 1.1.3. Questions must include “Not-applicable” option with score of 0
  - 1.2. The maximum score of the survey should be 100.
  - 1.3. The minimum score of the survey should be 0.
  - 1.4. The survey should have unlimited attempts.
2. The system must provide users with their score once the survey is completed.
3. The system must provide users with their risk assessment at the end of the survey.

- 3.1. If the score is 70 or higher: the user is within High Risk
  - 3.2. If the score is within the range 40 to 69: the user is within Medium risk
  - 3.3. If the score is 39 or lower: the user is within Low Risk
4. The system must be able to store the risk assessment and the score in the database under the User's name
5. The system must provide a list of reminders related to their risk level at the end of the survey.

## **4.6. Risk-reducing Measures**

### **4.6.1. Description and Priority**

This feature allows the users to refer to media and information regarding safety measures against mosquito breeding and dengue. This feature sends notifications to the users to complete the reminders based on the survey. This feature is of medium priority.

### **4.6.2. Stimulus/Response Sequences**

Users can click on the information icon button to review safety measures of different forms such as FAQ and educational videos. The user should complete the survey to enable notifications of reminders to be completed sent to the user.

### **4.6.3. Functional Requirements**

1. The system must provide information regarding safety measures in a new window.
2. The user must be able to select the type of information.
  - 2.1. System must provide information in the following formats:
    - 2.1.1. Video
    - 2.1.2. Text
  - 2.2. The user must be able to select the video option to review informative video related to dengue.
  - 2.3. The user must be able to access FAQ material.
3. The system must provide daily reminders.
  - 3.1. The system must provide daily notification to users based on the risk assessment from the survey the user completed.

- 3.2. Users must be able to select the notification and review list of measures to prevent dengue infestation within the users proximity.

## **4.7. Proximity Notification**

### **4.7.1. Description and Priority**

This feature notifies the user if:

1. His/her current GPS location is within 3 km of a known dengue cluster.

This feature is of high priority.

### **4.7.2. Stimulus/Response Sequences**

Users will provide the system with GPS tracking permissions. If there are known dengue clusters within 3 km of the user's current GPS location, the system will send a notification.

### **4.7.3. Functional Requirements**

1. The system must be able to access the user's GPS location data.
  - 1.1. The system must prompt the user to turn on GPS if it is not turned on.
  - 1.2. The system must prompt the user to provide the system with location permissions if it does not have it.
2. The system must notify users when there is a dengue cluster within 3 km of their current location in the form of in application notification.
3. The system must provide the following information in the application notifications:
  - 3.1. Name of cluster

## **5. Other Non-Functional Requirements**

### **5.1. Performance Requirements**

1. The system must not crash when the user opens the application.

2. The dengue clusters must be updated daily at 0100hrs.
  - 2.1. When the user prompts for dengue clusters, the system must return the latest real-time information within 45 seconds.
  - 2.2. When a user enters an area, given that they gave permission to allow the system to track their location, the system must notify the user of any clusters and number of cases if the area is within 3 km of a cluster, all of which must happen within 45 seconds of restarting the application.
3. System should provide users with a record of their previous dengue risk assessment and score until they retake the quiz.
4. The system must be able to maintain and run itself with little or no downtime incurred.

## **5.2. Security Requirements**

1. The system will mask the password field in order to prevent any potential shoulder surfing.
2. The system will have requirements set in place for passwords to ensure at least medium password strength.
3. The system will provide an OTP for additional security of the users' account.
4. The system must ensure that the user's privacy is not infringed.
  - 4.1. The users' credentials must be kept safe to protect the users' privacy.
  - 4.2. The users' search history must be kept safe to protect the user's privacy.
  - 4.3. The users' GPS location should be kept private to protect the user's privacy.

## **5.3. Software Quality Attributes**

1. The system must not be unavailable for more than 45 seconds.
2. The system must be capable of working with little supervision.
3. 80% of first-time users shall be able to complete registration within 5 minutes of starting to use the system

## **5.4. Reliability**

1. The system must be available 24 hours a day, 7 days a week.
2. After a system reboot, the full system functionality must be restored within 4 minutes.
3. The system must not crash 9 out of 10 times.

## **5.5. Usability**

1. The system must reduce short-term memory load.
  - 1.1. The User Interface must be simple and allow users to search for clusters easily.
  - 1.2. The User Interface must be simple and allow users to register and login easily.
2. The system must permit easy reversal of actions.
  - 2.1. The user must be able to retake the survey for Aedes risk-assessment easily.
  - 2.2. The user must be able to change the account password easily.
  - 2.3. The user must be able to add clusters to and remove clusters from their list of Saved Clusters easily.
3. The system must strive for consistency.
  - 3.1. A consistent sequence of actions is required for similar situations.
  - 3.2. The visual layout, such as buttons and labels, in the application must be consistent.



## 6. Other Requirements

### Appendix A: Glossary

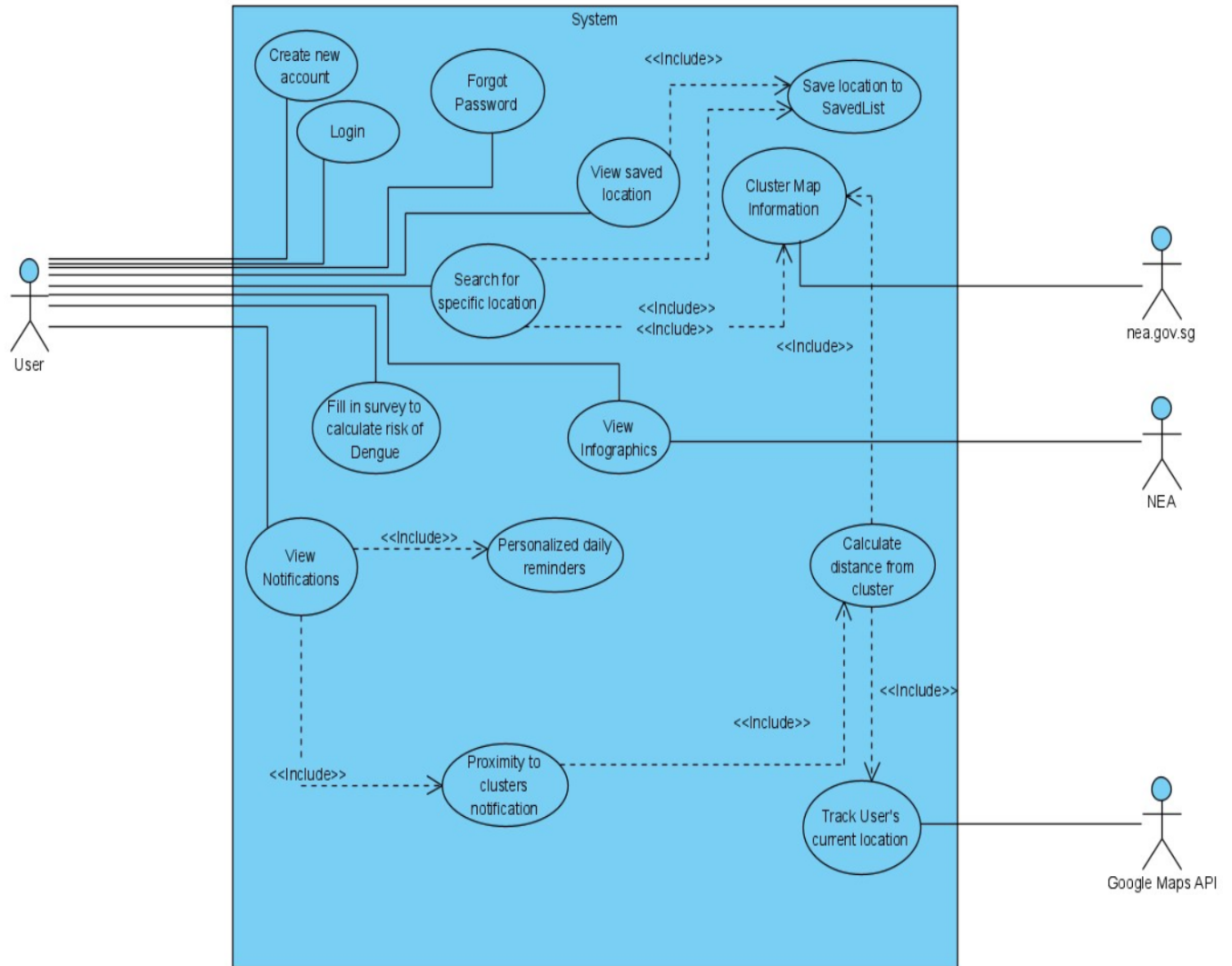
#### Data Dictionary

Term	Definition
<b>Aedes</b>	A genus ( <i>Aedes</i> ) of mosquitoes including the vector of yellow fever, dengue, and other diseases
<b>Dengue</b>	An acute infectious disease caused by a flavivirus (species <i>Dengue virus</i> of the genus <i>Flavivirus</i> ), transmitted by aedes mosquitoes, and characterised by headache, severe joint pain, and a rash
<b>Dengue Cluster</b>	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 (based on residential and workplace addresses)
<b>GPS</b>	Stands for global positioning system, an accurate worldwide navigational and surveying facility based on the reception of signals from an array of orbiting satellites
<b>Condominium</b>	A building or complex of buildings containing a number of individually owned apartments or houses.
<b>HDB</b>	Housing & Development Board (HDB) is Singapore's public housing authority. We plan and develop Singapore's housing estates; building homes and transforming towns to create a quality living environment for all

<b>Landed property</b>	Properties that are attached directly to the land that you purchase.
<b>System</b>	Our mobile application.
<b>Verify</b>	Check users' credentials with our database and make sure it matches
<b>User</b>	The person using our mobile application
<b>Guest User</b>	The person surfing our website/using our application but has no login details.
<b>NEA</b>	National Environment Agency is the leading public organisation responsible for ensuring a clean and green environment, and the sustainable development of Singapore
<b>Household</b>	A house and its occupants regarded as a unit
<b>Infestation</b>	The presence of an unusually large number of insects or animals in a place, typically so as to cause damage or disease

# Appendix B: Analysis Models

## 1. Use Case Diagram



## 2. Use Case Descriptions

### 2.1. User Registration

<b>Use Case ID:</b>	1		
<b>Use Case Name:</b>	Create New Account		
<b>Created By:</b>	Chen Xueyao	<b>Last Updated By:</b>	Wong Ze Ming
<b>Date Created:</b>	6 February 2021	<b>Date Last Updated:</b>	13 April 2021

<b>Actor:</b>	User
<b>Description:</b>	The user wants to register a new account.
<b>Preconditions:</b>	<ol style="list-style-type: none"><li>1. The user must already have downloaded the application onto their personal mobile device.</li><li>2. The user must have access to the internet.</li><li>3. The user must have an email which they can receive emails from.</li><li>4. User account must not already exist in database</li></ol>
<b>Postconditions:</b>	<ol style="list-style-type: none"><li>1. User account is successfully registered in the database.</li></ol>
<b>Priority:</b>	Medium
<b>Frequency of Use:</b>	Approximately 1-2 times
<b>Flow of Events:</b>	<p>Create new account:</p> <ol style="list-style-type: none"><li>1. User chooses the 'Sign Up' option in the main menu of the application.</li><li>2. The user then fills up the email address field.</li><li>3. The system sends a verification email to the user's email with an OTP.</li><li>4. The user verifies his/her account by entering the OTP and the application checks that it matches.</li></ol>

	5. User then fills in the username, email address and password fields. 6. The system verifies the fields for accuracy of input. 7. The account is then created.		
<b>Alternative Flows:</b>	Username already taken: <ol style="list-style-type: none"> <li>1. The user fills in ‘username’, ‘email’ and ‘password’ fields in the sign-up menu.</li> <li>2. The system checks input.</li> <li>3. The system alerts the user that his/her username is already taken.</li> <li>4. The system prompts the user to re-enter a new username.</li> </ol>		
<b>Exception:</b>	Signing up when the user already has an account: <ol style="list-style-type: none"> <li>1. User fills in ‘username’, ‘email’ and ‘password’ fields in the sign-up menu.</li> <li>2. The system checks input. The email address already exists in the system database.</li> <li>3. The system prompts the user that he/she already has an account.</li> <li>4. The system re-directs the user to the log-in menu.</li> </ol>		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	-		
<b>Notes and Issues:</b>	-		

## 2.2. User Login

<b>Use Case ID:</b>	2		
<b>Use Case Name:</b>	Login		
<b>Created By:</b>	Chen Xueyao	<b>Last Updated By:</b>	Wong Ze Ming

<b>Date Created:</b>	6 February 2021	<b>Date Last Updated:</b>	13 April 2021
----------------------	-----------------	---------------------------	---------------

<b>Actor:</b>	User
<b>Description:</b>	The user wants to login to their registered account
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The user must already have downloaded the application onto their personal mobile device.</li> <li>2. The user must have access to the internet.</li> <li>3. The user must have an email which they can receive emails from.</li> </ol>
<b>Postconditions</b>	The user is able to successfully login to their registered account
<b>Priority:</b>	High
<b>Frequency of Use:</b>	Very frequently
<b>Flow of Events:</b>	<p>Login:</p> <ol style="list-style-type: none"> <li>1. The user inputs his/her username and password.</li> <li>2. The system verifies for accuracy of input by checking the user's credentials with the system's database.</li> <li>3. The system authenticates the user to login to the application successfully.</li> </ol>
<b>Alternative Flows:</b>	
<b>Exception:</b>	<ol style="list-style-type: none"> <li>1. Logging in when the user does not have an account:</li> <li>2. User fills in the 'username' and 'password' field in the login menu.</li> <li>3. The system checks input. The username does not exist in the system database.</li> <li>4. The system alerts the user that he/she does not have an account.</li> <li>5. The system prompts the user to create an account.</li> <li>6. The system redirects the user to the welcome page.</li> <li>7. The user selects the 'Sign Up' option.</li> </ol>
<b>Includes:</b>	-

<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

## 2.3. Forgot Password

<b>Use Case ID:</b>	3		
<b>Use Case Name:</b>	Forgot Password		
<b>Created By:</b>	Wong Ze Ming	<b>Last Updated By:</b>	
<b>Date Created:</b>	13 April 2021	<b>Date Last Updated:</b>	

<b>Actor:</b>	User
<b>Description:</b>	The user forgets his/her password and wants to change it
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. The user must already have downloaded the application onto their personal mobile device.</li> <li>2. The user must have access to the internet.</li> <li>3. The user must have already created an account and it is registered in our database.</li> </ol>
<b>Postconditions</b>	The user successfully changes his/her password
<b>Priority:</b>	High
<b>Frequency of Use:</b>	Infrequently
<b>Flow of Events:</b>	<p>Forgot password:</p> <ol style="list-style-type: none"> <li>1. The user selects the 'Forgot Password' option on the log-in menu.</li> </ol>

	<ol style="list-style-type: none"> <li>2. The user inputs the email address into the field.</li> <li>3. The system checks if the email address is valid.</li> <li>4. If valid, the system sends a verification email to the user.</li> <li>5. The user accesses the link sent within the verification email.</li> <li>6. The system prompts the user for a new password.</li> <li>7. The user inputs his/her new password.</li> <li>8. The system will feedback to the user that his/her password has been updated.</li> <li>9. The new password will be updated into the database and the password for the user is changed.</li> </ol>
<b>Alternative Flows:</b>	-
<b>Exception:</b>	-
<b>Includes:</b>	-
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

## 2.4. Dengue Risk Calculation

<b>Use Case ID:</b>	4		
<b>Use Case Name:</b>	Fill in survey to calculate risk of dengue		
<b>Created By:</b>	Chen Xueyao	<b>Last Updated By:</b>	Wong Ze Ming
<b>Date Created:</b>	7 February 2021	<b>Date Last Updated:</b>	13 April 2021



<b>Actor:</b>	User
<b>Description:</b>	The user calculates his/her risk of dengue by filling in a survey. The system will provide a score, out of 100 with 100 being the highest risk, to the user.
<b>Preconditions:</b>	1. The user must be logged in to the application.
<b>Postconditions:</b>	The system displays the survey score on the screen.
<b>Priority:</b>	High
<b>Frequency of Use:</b>	Approximately once a week
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. The user clicks on the survey icon at the bottom of the display.</li> <li>2. The user fills in each question of the survey.</li> <li>3. The system checks for validity of user input.</li> <li>4. Once the survey is completed, the system calculates and displays the total score for the user as well as the risk level attributed to that score.</li> <li>5. The system displays a link which the user can click to enable daily reminders</li> </ol>
<b>Alternative Flows:</b>	View risk reducing measures: <ol style="list-style-type: none"> <li>1. User clicks on the link provided by the system at the end of the quiz.</li> <li>2. User is redirected to the Safety Measures page.</li> </ol>
<b>Exception:</b>	-
<b>Includes:</b>	Reminders
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

## 2.5. Proximity Notification

<b>Use Case ID:</b>	5		
<b>Use Case Name:</b>	Proximity Notification		
<b>Created By:</b>	Wong Ze Ming	<b>Last Updated By:</b>	Wong Ze Ming
<b>Date Created:</b>	6th February 2021	<b>Date Last Updated:</b>	13 April 2021

<b>Actor:</b>	Google_map_api, User, NEA.gov.sg
<b>Description:</b>	Live GPS tracking and notification of cluster proximity
<b>Preconditions:</b>	<ol style="list-style-type: none"><li>1. User account is already in the database</li><li>2. User's phone must be connected to WIFI/Mobile Data</li><li>3. User's phone must have GPS turned on</li><li>4. User must allow our application to have access to his/her location</li></ol>
<b>Postconditions:</b>	The system is able to track the user's current location and send notifications to the user when he/she travels within 3km of a dengue cluster.
<b>Priority:</b>	High
<b>Frequency of Use:</b>	0-10 times per day
<b>Flow of Events:</b>	<ol style="list-style-type: none"><li>1. The user travels within 3km of a known dengue cluster</li><li>2. The system picks this up using GPS tracker</li><li>3. The application sends a notification to the user about his/her proximity to the cluster inside the application.</li><li>4. The notification contains the name of cluster, number of cases in the cluster and distance of cluster to user's current location</li></ol>
<b>Alternative Flows:</b>	-
<b>Exception:</b>	<ol style="list-style-type: none"><li>1. User has not turned on GPS<ol style="list-style-type: none"><li>1.1. The system will prompt the user with a notification to turn</li></ol></li></ol>

	<p>on GPS</p> <ol style="list-style-type: none"> <li>2. User has not given the system location permissions <ol style="list-style-type: none"> <li>2.1. The system will prompt the user with a notification to provide location permissions to the application</li> </ol> </li> </ol>
<b>Includes:</b>	-
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	<ol style="list-style-type: none"> <li>1. Assumes user has provided the system with GPS tracking permission</li> </ol>
<b>Notes and Issues:</b>	-

## 2.6. Safety Measures

<b>Use Case ID:</b>	6		
<b>Use Case Name:</b>	Safety Measures		
<b>Created By:</b>	Palaniselvam Shyam Sundar	<b>Last Updated By:</b>	Wong Ze Ming
<b>Date Created:</b>	6th February 2021	<b>Date Last Updated:</b>	13 April 2021

<b>Actor:</b>	User, NEA
<b>Description:</b>	A list of daily reminders will be provided by the System, which can also be enabled to be sent in the form of daily notifications to aid users.
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User account is already in the database</li> <li>2. User's phone must be connected to WIFI/Mobile Data</li> </ol>

	<ol style="list-style-type: none"> <li>3. User's phone must have GPS turned on</li> <li>4. User must allow our application to have access to his/her location</li> <li>5. Users must allow the application to send notifications on his/her device.</li> <li>6. Users must have completed the survey and gotten a risk-assessment</li> </ol>
<b>Postconditions:</b>	The system displays a list of measures to reduce the risk of dengue in the user's household, as well as sending daily reminders to the user
<b>Priority:</b>	Medium
<b>Frequency of Use:</b>	Daily usage
<b>Flow of Events:</b>	<p><u>Safety measures</u></p> <ol style="list-style-type: none"> <li>1. Option provided by the System to view reminders</li> <li>2. Upon User choosing that option, a new window is opened with a list of reminders that he/she can do to reduce the risk of dengue. The reminders are catered according to the risk-assessed through the survey mentioned earlier.</li> </ol> <p><u>Daily reminders</u></p> <ol style="list-style-type: none"> <li>1. User enables daily reminders after completing the survey</li> <li>1. The system then will provide the user with daily reminders with safety measures based on their risk-assessment for the survey.</li> <li>2. User will be able to tap the notification and review a list of recommended measures to take to reduce the risk of Dengue.</li> </ol>
<b>Alternative Flows:</b>	-
<b>Exception:</b>	-
<b>Includes:</b>	-
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-

## 2.7. Map of Clusters

<b>Use Case ID:</b>	7		
<b>Use Case Name:</b>	Map of Dengue Clusters		
<b>Created By:</b>	Palaniselvam Shyam Sundar	<b>Last Updated By:</b>	Wang Kangxin
<b>Date Created:</b>	6th February 2021	<b>Date Last Updated:</b>	16 April 2021

<b>Actor:</b>	NEA.gov.sg, user, google_map_api
<b>Description:</b>	The System will show the user an interactive map of known Dengue clusters (represented in red). This map will be updated every 4 hours in order to keep the users better informed and avoid entering known clusters.
<b>Preconditions:</b>	<ol style="list-style-type: none"><li>1. User account is already in the database</li><li>2. User's phone must be connected to WIFI/Mobile Data</li><li>3. User's phone must have GPS turned on</li><li>4. User must allow our application to have access to his/her location</li></ol>
<b>Postconditions</b>	Interactive map is displayed.
<b>Priority:</b>	High
<b>Frequency of Use:</b>	Daily
<b>Flow of Events:</b>	<ol style="list-style-type: none"><li>1. System will display an interactive map of known Dengue clusters (updated every 4 hours according to NEA.gov.sg).</li><li>2. Dengue clusters will be represented in red.</li><li>3. Users will then interact with the map through zoom in, zoom out, drag and rotate functions.</li><li>4. User clicks on a cluster, displaying its name, current number of dengue cases, level of risk and cases since the start of the cluster's inception.</li></ol>

<b>Alternative Flows:</b>	
<b>Exception:</b>	
<b>Includes:</b>	-
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	NEA has agreed to give daily updates and reports to our system.
<b>Notes and Issues:</b>	-

## 2.8. Saving of location

<b>Use Case ID:</b>	8		
<b>Use Case Name:</b>	Saving of location		
<b>Created By:</b>	Wong Ze Ming	<b>Last Updated By:</b>	
<b>Date Created:</b>	17 April 2021	<b>Date Last Updated:</b>	

<b>Actor:</b>	Google_map_api, User, NEA.gov.sg
<b>Description:</b>	Live GPS tracking and notification of cluster proximity
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User account is already in the database</li> <li>2. User's phone must be connected to WIFI/Mobile Data</li> <li>3. User's phone must have GPS turned on</li> <li>4. User must allow our application to have access to his/her location</li> </ol>
<b>Postconditions:</b>	The system saves the dengue that the user selected into his/her "Saved Locations" and the database is updated accordingly
<b>Priority:</b>	Medium

<b>Frequency of Use:</b>	0-3 times per day		
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. The user searches for a location in Singapore using our application</li> <li>2. He/she saves the location to his/her application</li> <li>3. If the saved location is within 3km radius of a dengue cluster, the system sends a notification to the user</li> <li>4. The notification contains the name of cluster, number of cases in the cluster, the risk level and distance of cluster to user's saved location</li> </ol>		
<b>Alternative Flows:</b>	-		
<b>Exception:</b>	<ol style="list-style-type: none"> <li>3. User has not turned on GPS               <ol style="list-style-type: none"> <li>3.1. The system will prompt the user with a notification to turn on GPS</li> </ol> </li> <li>4. User has not given the system location permissions               <ol style="list-style-type: none"> <li>4.1. The system will prompt the user with a notification to provide location permissions to the application</li> </ol> </li> </ol>		
<b>Includes:</b>	-		
<b>Special Requirements:</b>	-		
<b>Assumptions:</b>	<ol style="list-style-type: none"> <li>2. Assumes user has provided the system with GPS tracking permission</li> </ol>		
<b>Notes and Issues:</b>	-		

## 2.9. View Infographics

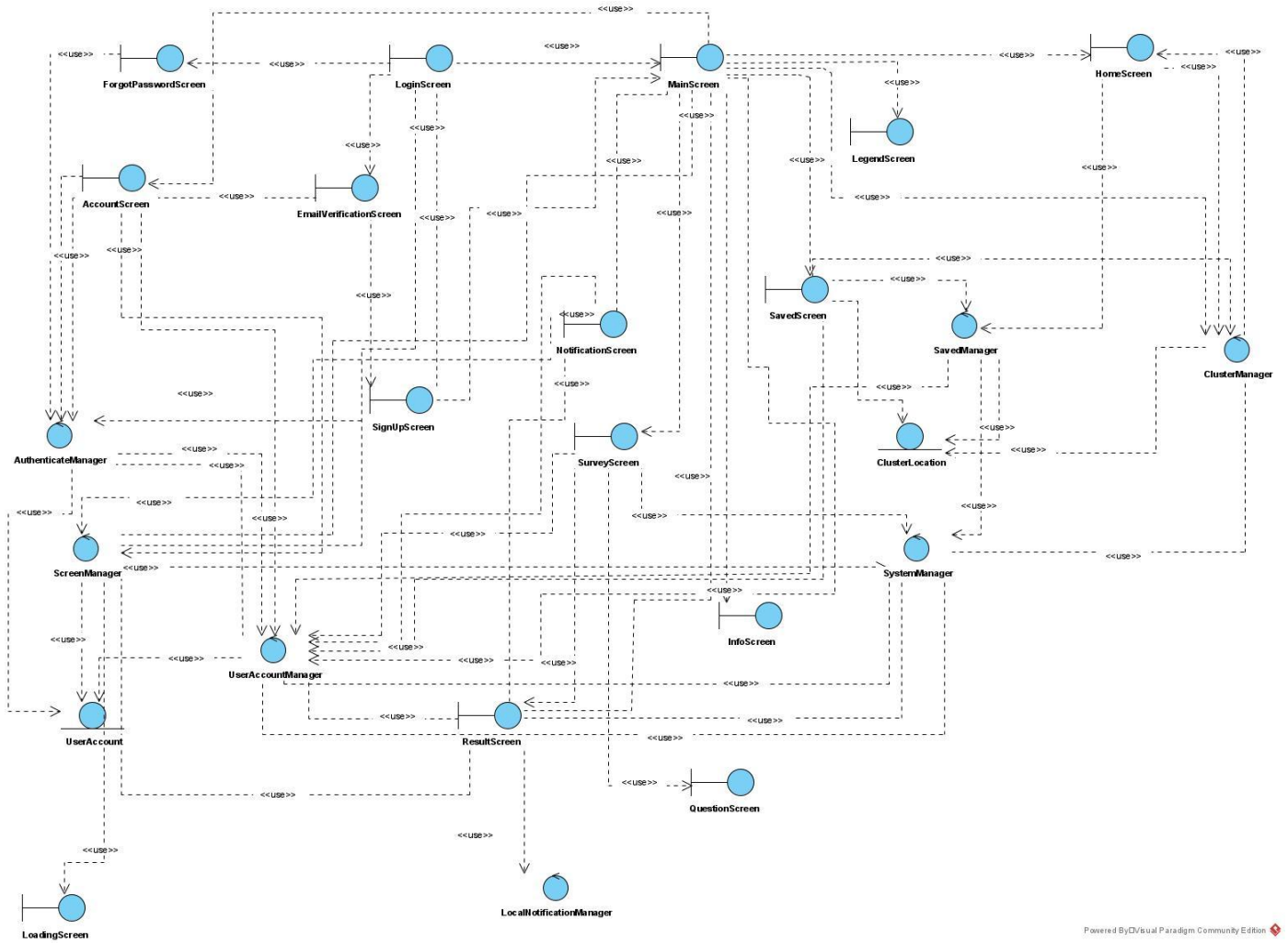
<b>Use Case ID:</b>	9		
<b>Use Case Name:</b>	View Infographics		
<b>Created By:</b>	Wong Ze Ming	<b>Last Updated By:</b>	

<b>Date Created:</b>	17 April 2021	<b>Date Last Updated:</b>	
----------------------	---------------	---------------------------	--

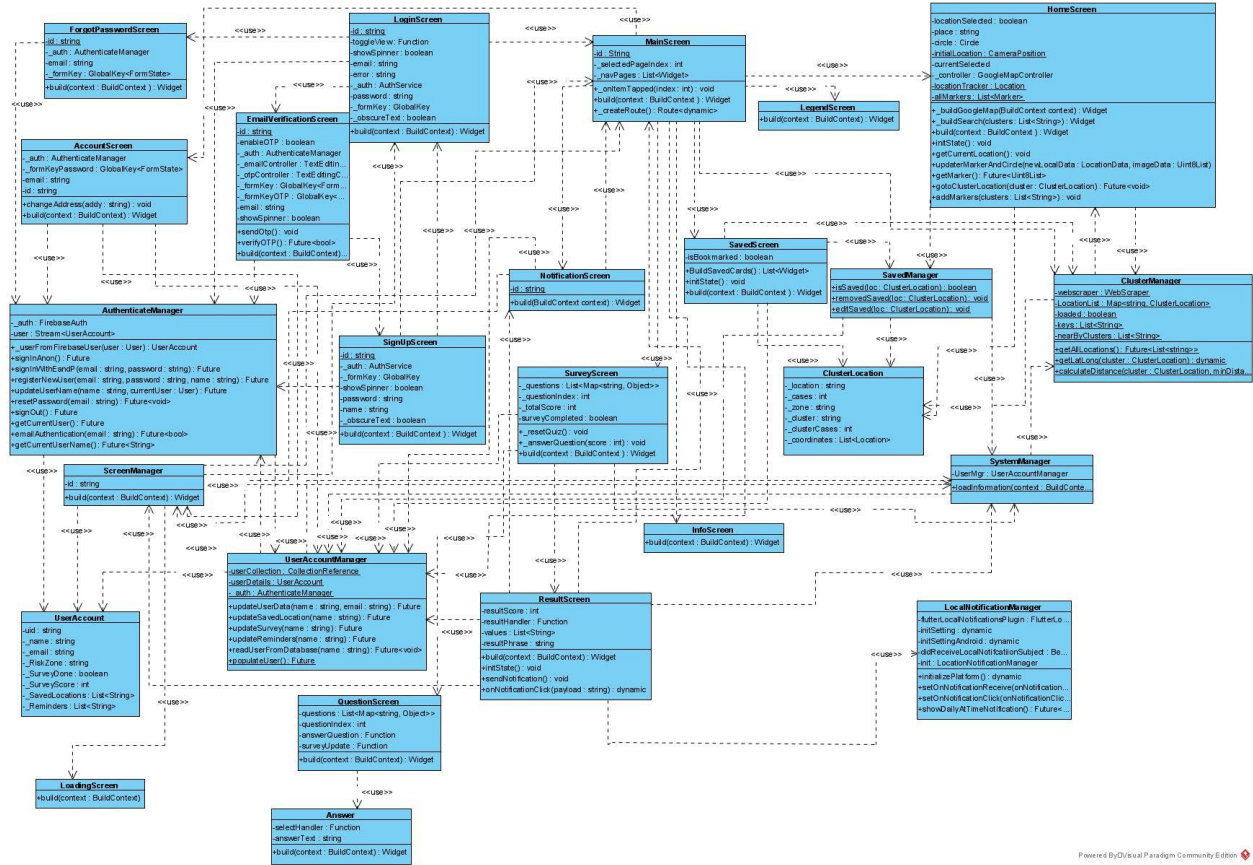
<b>Actor:</b>	User, NEA
<b>Description:</b>	View information on how to prevent dengue
<b>Preconditions:</b>	<ol style="list-style-type: none"> <li>1. User account is already in the database</li> <li>2. User's phone must be connected to WIFI/Mobile Data</li> </ol>
<b>Postconditions:</b>	User views the information to combat dengue
<b>Priority:</b>	Low
<b>Frequency of Use:</b>	0-1 time per day
<b>Flow of Events:</b>	<ol style="list-style-type: none"> <li>1. The user taps on the view information button</li> <li>2. A list of information in the form of text and video is presented for his/her viewing</li> </ol>
<b>Alternative Flows:</b>	-
<b>Exception:</b>	-
<b>Includes:</b>	-
<b>Special Requirements:</b>	-
<b>Assumptions:</b>	-
<b>Notes and Issues:</b>	-



### 3. Conceptual Class Diagram

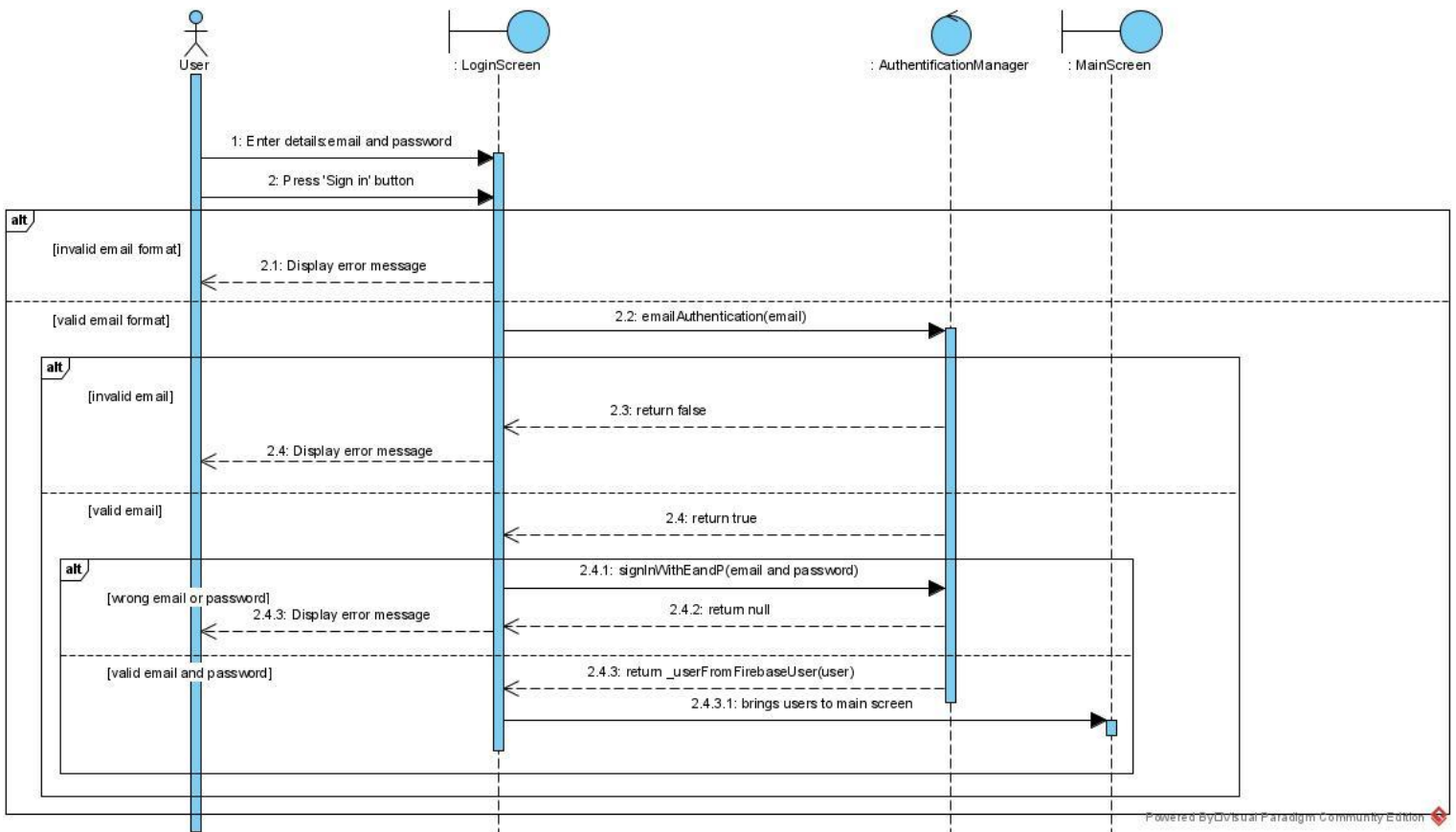


## 4. Final Class Diagram

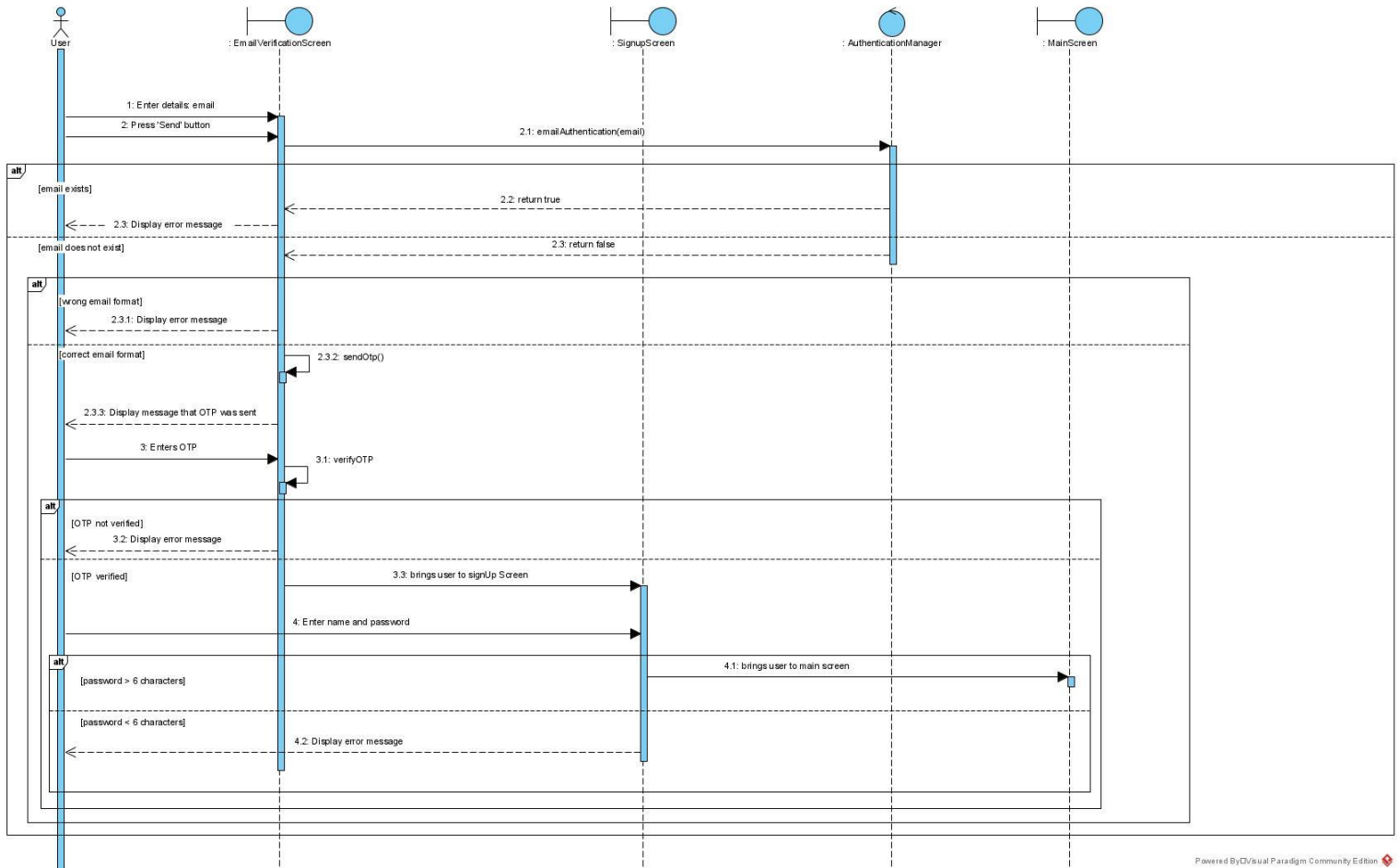


## 4. Sequence Diagrams

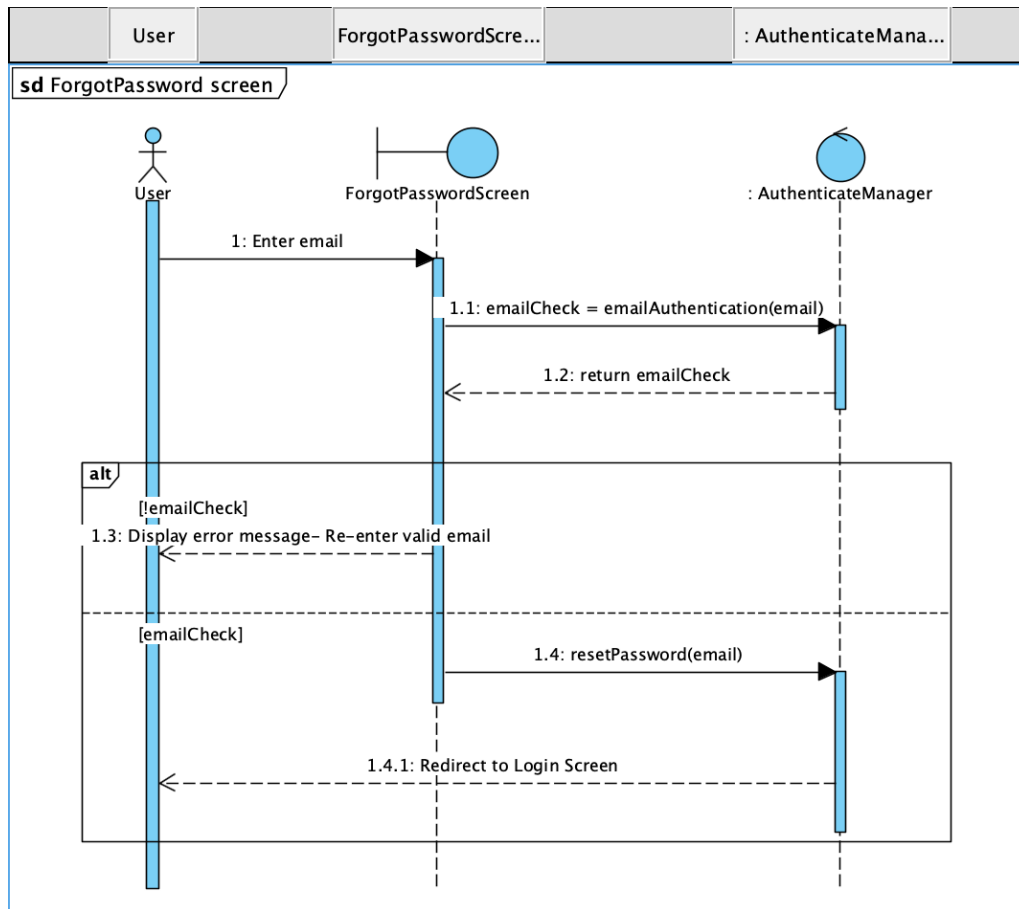
### 1. Login



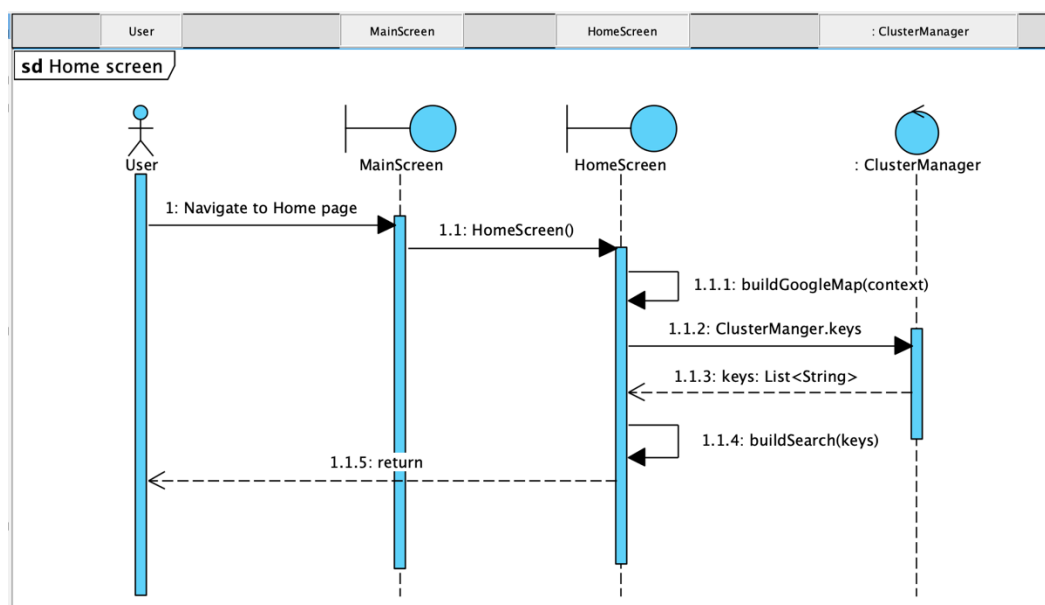
## 2. Register



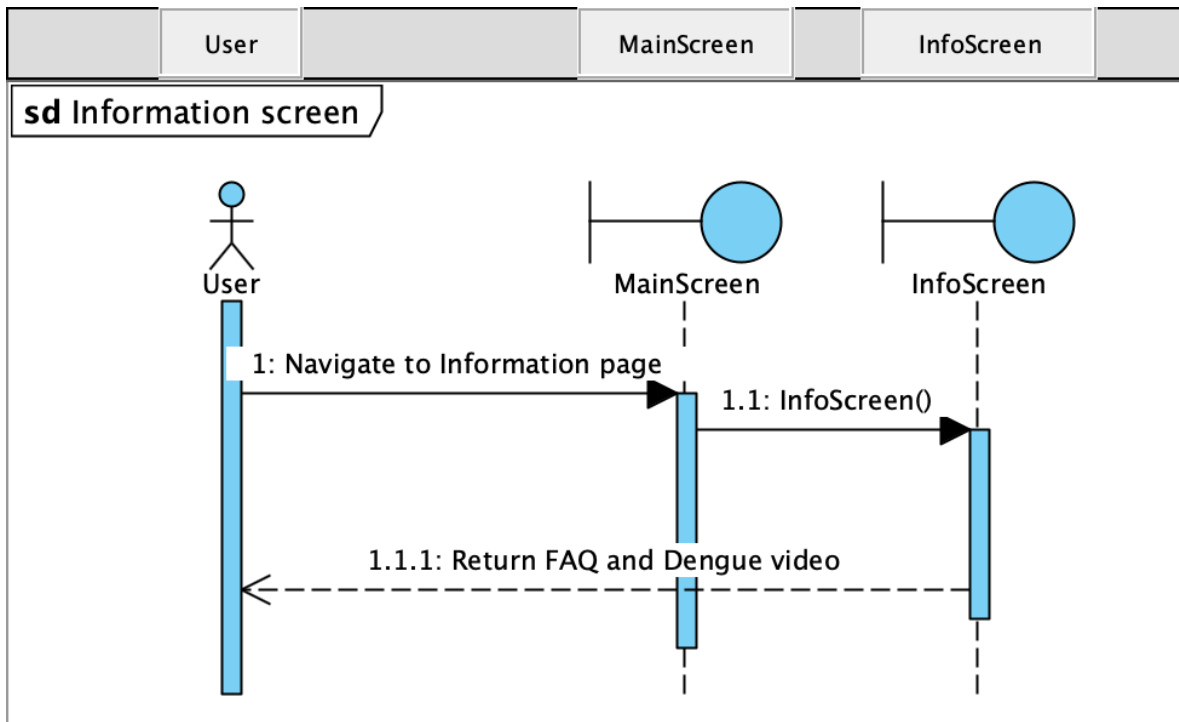
### 3. Forgot password



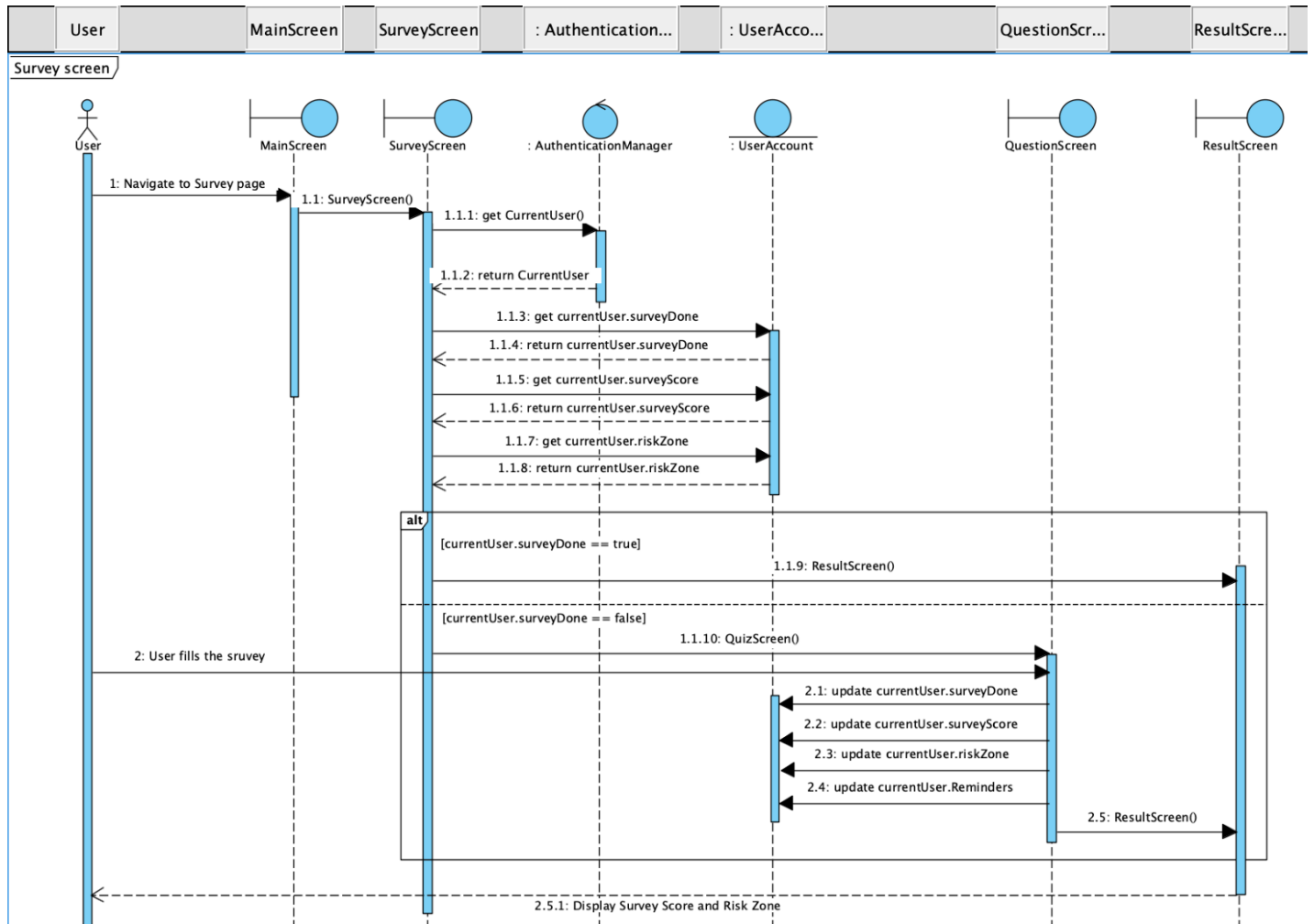
### 4. Home Screen



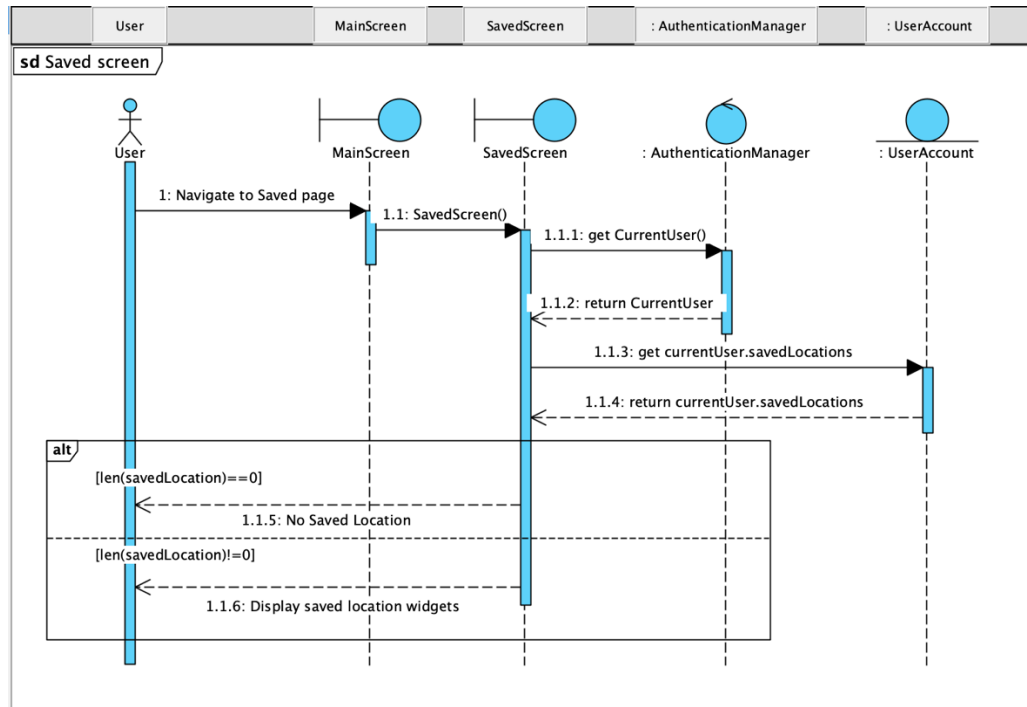
## 5. Information Screen



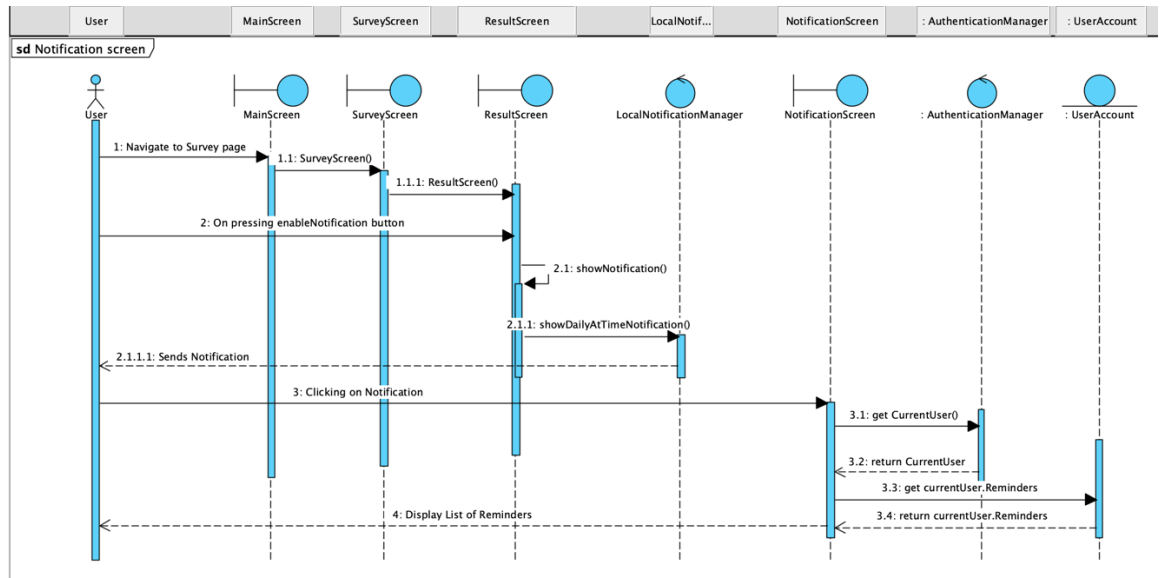
## 6. Survey Screen



## 7. Saved Location Screen

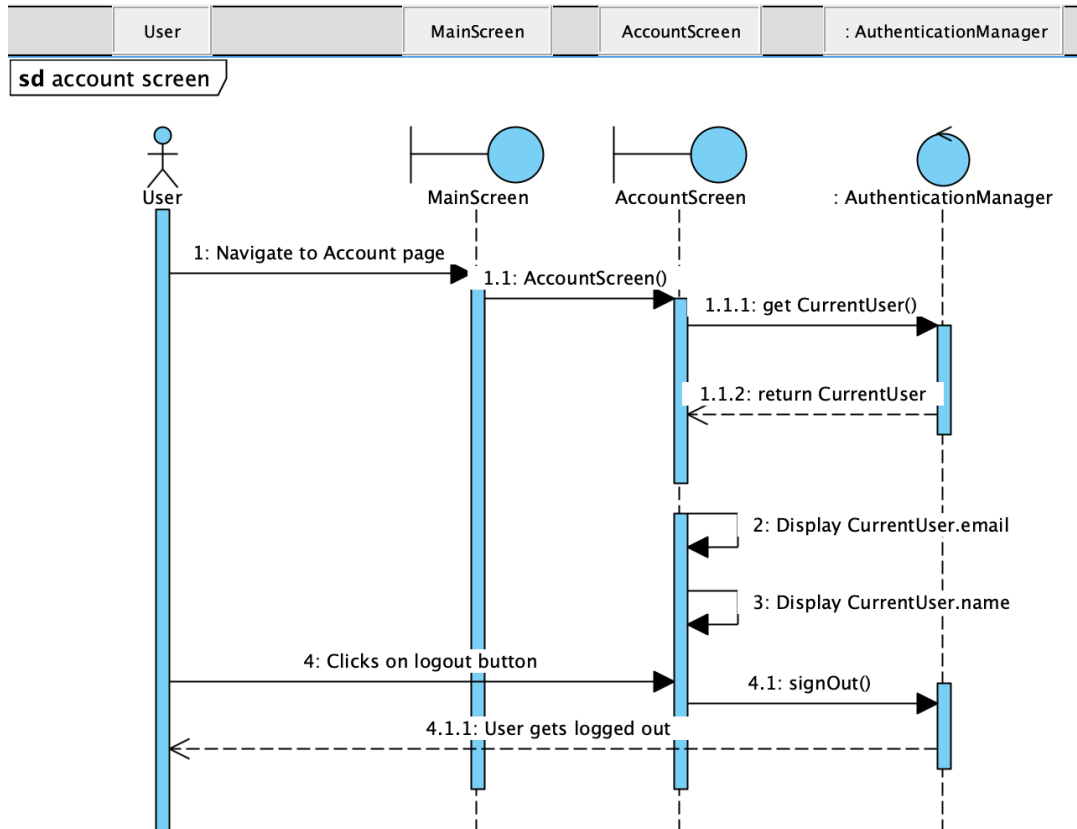


## 8. Notification Screen

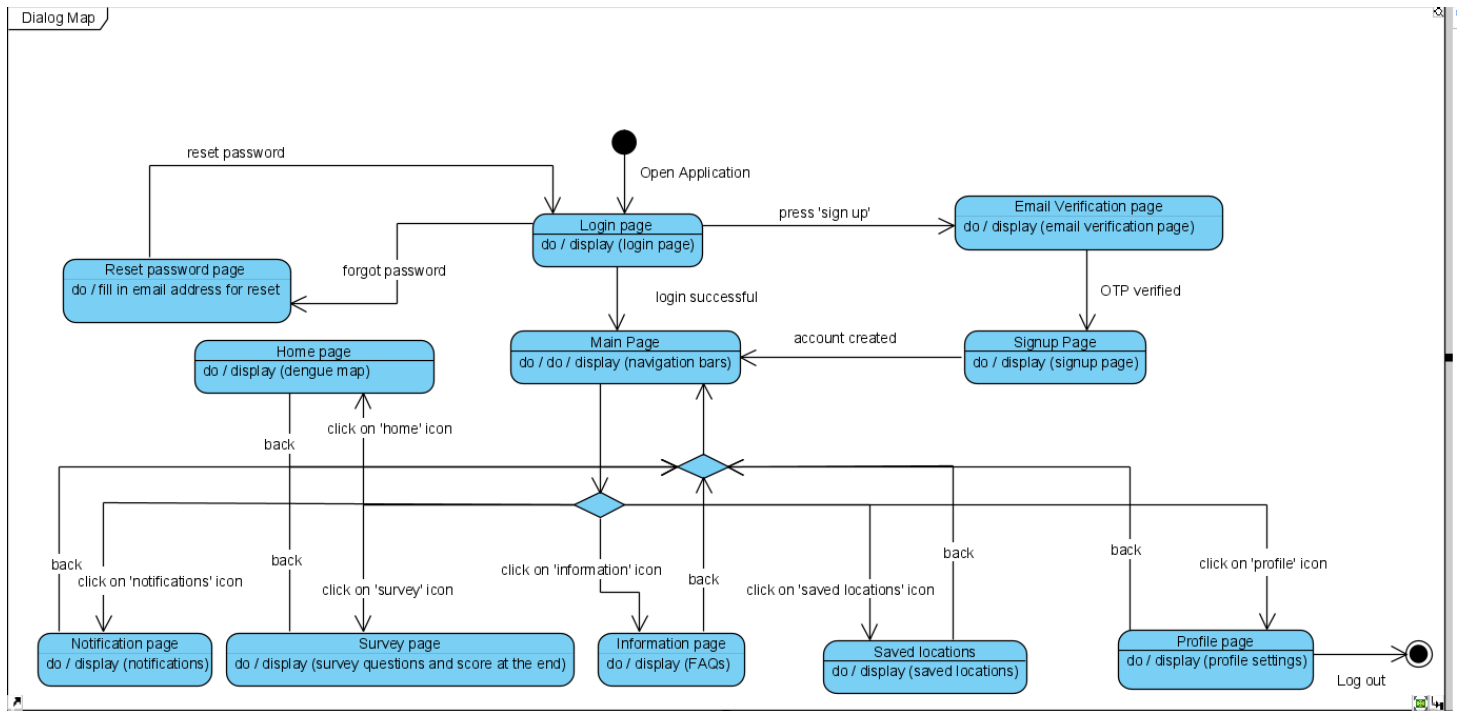




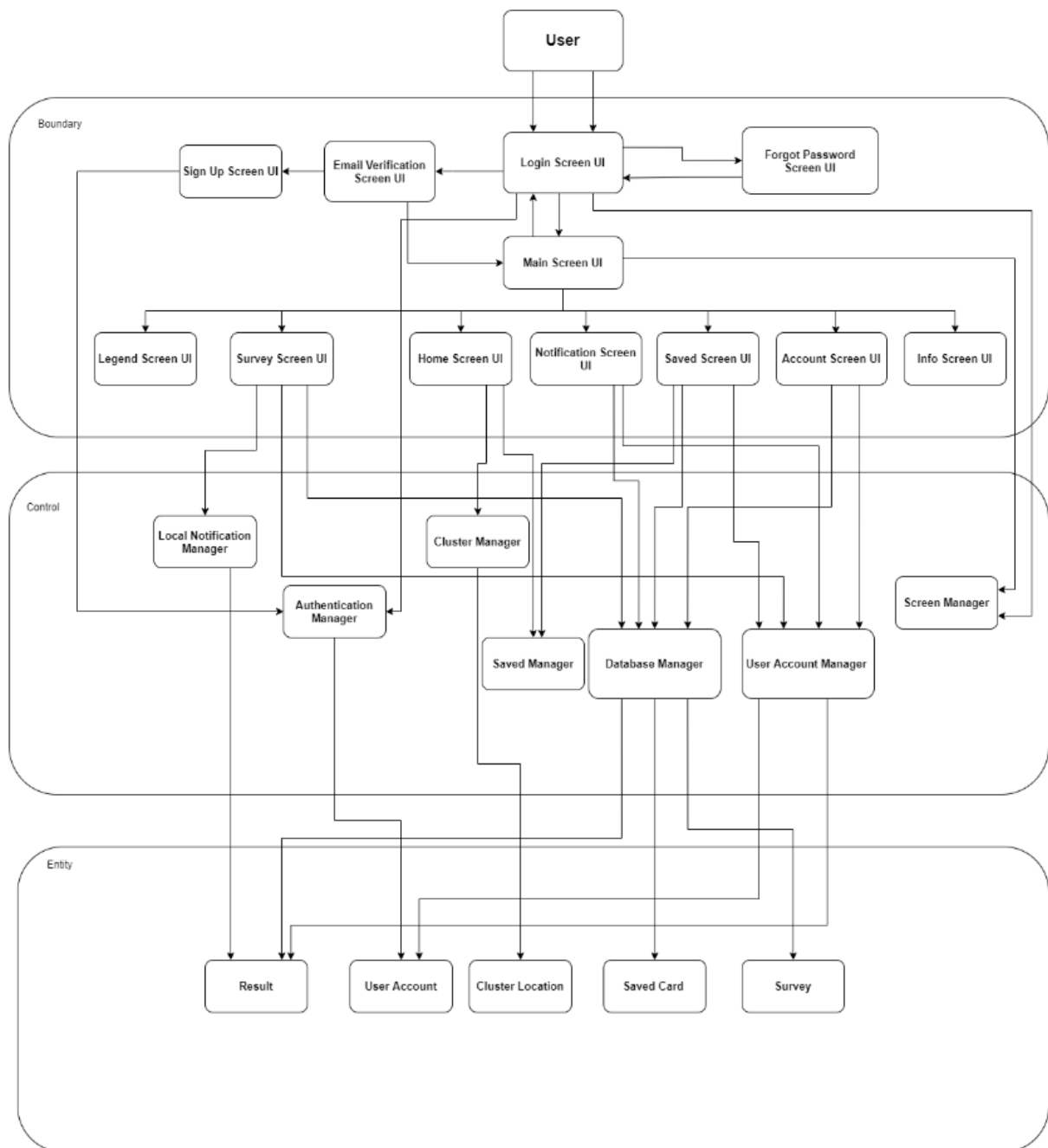
## 9. Account Screen



## 5. Dialog Map



## 6. System Architecture and Design



## **6.1. Architecture**

This program is implemented in the Layered Architecture format.

Rationale for selection:

- 1) The architecture allows segregation of the program into departments where each layer is concerned with (and looks after) its set of functions, processes and controls.
- 2) This type of architecture provides added security as well. Only the Boundary layer is visible to the outside world and the other two layers are abstracted. In order to compromise the system, the attacker would have to go through many layers before getting to something worth hacking.
- 3) Simplicity: this architecture is easier to implement
- 4) Consistency: this architecture allows better and more consistent organization of code throughout
- 5) The architecture allows parallel working and this is especially beneficial when working in teams. Members can work on separate parts of the architecture with minimal dependencies on other parts.

## **6.2. Design Patterns**

Strategy, Factory and Observer are the design patterns that have been taken into consideration.

The Factory pattern is seen in the use of the different entity classes. For instance, the UserAccount entity class allows for the creation of separate users (as objects) and each user is then able to interact with the control class separately. Control classes are accessed by the Boundary classes and as a result, loose coupling is enabled.

The Boundary classes are concerned with the entire UI of the application and enable easy changes, even during runtime. This also prevents the UI from interfering with any functionality of the application. Observer pattern is visible here as there is loose coupling between the UI, functionality and data of the application.

The Control classes implement the functionality within the application and interact with the Entity classes as well. Strategy patterns can be observed using this structure.

## 7. Testing

### 1. Black Box Testing

#### 1.1. Sign In

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	All valid inputs	Email and password keyed in correspond to an account in the database.	The system will log user into the app, Main Screen displayed	System will log user into the app, Main Screen displayed
2	No input in 'Email' field	User enters a password but not email.	The system will display an error message, prompting the user to 'Enter email'.	The system will display an error message, prompting the user to 'Enter email'.
3	Invalid input in 'Email' field (not found in database)	User's email input does not exist in database	<p>The system will display an error message indicating that the account does not exist.</p> <p>The system will prompt for the user to log in to a different account or register for a new account.</p>	<p>The system will display an error message indicating that the account does not exist.</p> <p>The system will prompt for the user to log in to a different account or register for a new account.</p>
4	Invalid input in 'Email' field (input is not an	User's input in the 'Email' field is not in the format of an	The system will indicate to the user that the email format is invalid.	The system will indicate to the user that the email format is invalid.

	email)	email.  Eg.  “ <b>abc</b> ” instead of “abc@gmail.com”		
5	No input in ‘Password’ Field	The user inputs an email but does not provide their password.	The system will display an error message prompting the user to re-enter their password.	The system will display an error message prompting the user to re-enter their password.
6	Incorrect input in ‘Password’ field (not found in database)	User’s password input does not correspond to the entry in the database.	The system will display an error message indicating invalid email and password.  The system will prompt for the user to re-enter their log-in credentials.	The system will display an error message indicating invalid email and password.  The system will prompt for the user to re-enter their log-in credentials..
7	No input in either field	The user presses the ‘Sign in’ button without filling in both email and password fields.	The system will display two error messages prompting the user to enter email and to enter password.	The system will display two error messages prompting the user to enter email and to enter password.

## 1.2. Verify OTP

Test ID	Test Name	Scenario	Expected Result	Actual Result
---------	-----------	----------	-----------------	---------------

1	Valid email	User inputs a valid email when registering for an account	The system will send an OTP to the provided email. The system will inform the user that an email has been sent and prompt the user to check their email.	The system will send an OTP to the provided email. The system will inform the user that an email has been sent and prompt the user to check their email.
2	No email input	User does not input an email before tapping on the 'Get OTP' button.	The system will display an error message informing the user that the email format is invalid.	The system will display an error message informing the user that the email format is invalid.
3	Invalid email (wrong format)	User input for the 'email' field does not follow the format of an email.  Eg.  “ <b>abc</b> ” instead of “abc@gmail.com”	The system will display an error message informing the user that the email format is invalid.	The system will display an error message informing the user that the email format is invalid.
4	Invalid email (email already exists)	User inputs an email that has already been linked to an existing account	The system will display an error message informing the user that the email already exists. The system will prompt the user to enter a separate email.	The system will display an error message informing the user that the email already exists. The system will prompt the user to enter a separate email.



4	Valid OTP	The user inputs a valid OTP into the 'OTP' field.	The system will go to the next screen, where users can fill in their name and password.	The system will go to the next screen, where users can fill in their name and password.
5	Invalid OTP	The user inputs an invalid OTP into the 'OTP' field.	The system will display an error message informing user that OTP is invalid and prompt for user to re-enter OTP	The system will display an error message informing user that OTP is invalid and prompt for user to re-enter OTP
6	No OTP	The user does not input any OTP into the 'OTP' field.	The system will display an error message prompting the user to 'enter OTP'.	The system will display an error message prompting the user to 'enter OTP'.

### 1.3. Register account

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	All valid inputs	User inputs valid name and password in the registration screen	The system creates the account and redirects users to the app's Home Screen.	The system creates the account and redirects users to the app's Home Screen.
2	No name provided	User inputs a password but does not provide a name.	The system will display an error message prompting the user to enter a name.	The system will display an error message prompting the user to enter a name.
4	No password provided	The user does not provide a password in the 'Password' field in the registration screen	The system will display an error message prompting the user to enter a password with 6 or more characters.	The system will display an error message prompting the user to enter a password with 6 or more characters.
5	Invalid password	The user inputs an invalid password in the registration screen	The system will display an error message prompting the user to input a password with 6 or more characters	The system will display an error message prompting the user to input a password with 6 or more characters

#### 1.4. Forgot Password

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	Valid email	The email provided is a valid email linked to an account in the database.	The system sends an email to the registered user's email containing a link for resetting the password. At the same time, within the app, the user is redirected to the sign-in screen.	The system sends an email to the registered user's email containing a link for resetting the password. At the same time, within the app, the user is redirected to the sign-in screen.
2	No email provided	The user does not provide an email in the 'Forgot Password' screen before clicking 'Submit'	The system will display an error message prompting the user to 'Enter email'.	The system will display an error message prompting the user to 'Enter email'.
3	Invalid Email format	The user input in the 'email' field does not follow the format of an email.  Eg.  "abc" instead of "abc@gmail.com"	The system will display an error message indicating to the user that their input is of an invalid email format.	The system will display an error message indicating to the user that their input is of an invalid email format.
4	Email does not exist in system	User input for email does not exist in the database (ie. the email is not linked to any account in the	The system indicates to the user that the email does not exist in the system and prompts the user to try a	The system indicates to the user that the email does not exist in the system and prompts the user to try a

		database).	new email.	new email.
5	Valid password used for reset	The user uses the link sent to their email to reset their password. They input a valid password.	The system indicates to the user that their password has been changed and they can now sign in with their new password.	The system indicates to the user that their password has been changed and they can now sign in with their new password.
6	No password provided	The user uses the link sent to their email to reset their password.  They leave the new password field blank.	The system displays an error message prompting the user to enter a password.	The system displays an error message prompting the user to enter a password.
7	Invalid password used for reset	The user uses the link sent to their email to reset their password. They input an invalid password.	The system indicates to the user that the password they provided is invalid.  The system prompts the user to create a strong password containing at least 6 characters.	The system indicates to the user that the password they provided is invalid.  The system prompts the user to create a strong password containing at least 6 characters.

### 1.5. Change Password

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	Valid email	The user provides the correct email linked to	The system sends an email to the registered	The system sends an email to the registered

	input	their DengueGo account (displayed in the same screen).	user's email containing a link for resetting the password.	user's email containing a link for resetting the password.
2	Valid password used for reset	The user uses the link sent to their email to reset their password. They input a valid password.	The system indicates to the user that their password has been changed and they can now sign in with their new password.	The system indicates to the user that their password has been changed and they can now sign in with their new password.
3	No password provided	The user uses the link sent to their email to reset their password.  They leave the new password field blank.	The system displays an error message prompting the user to enter a password.	The system displays an error message prompting the user to enter a password.
4	Invalid password used for reset	The user uses the link sent to their email to reset their password. They input an invalid password.	The system indicates to the user that the password they provided is invalid.  The system prompts the user to create a strong password containing at least 6 characters.	The system indicates to the user that the password they provided is invalid.  The system prompts the user to create a strong password containing at least 6 characters.

### 1.6. Survey

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	Low risk	User's score for the survey is in the 0-39 range.	The system shows the user their risk score out of 100 and indicates to them that they are at low risk of contracting dengue.	The system shows the user their risk score out of 100 and indicates to them that they are at low risk of contracting dengue.
2	Medium risk	User's score for the survey is in the 40-69 range.	The system shows the user their risk score out of 100 and indicates to them that they are at medium risk of contracting dengue.	The system shows the user their risk score out of 100 and indicates to them that they are at medium risk of contracting dengue.
3	High risk	User's score for the survey is in the 70-100 range.	The system shows the user their risk score out of 100 and indicates to them that they are at high risk of contracting dengue.	The system shows the user their risk score out of 100 and indicates to them that they are at high risk of contracting dengue.

### 1.7. Cluster Information

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	Low risk dengue cluster	User selects a low-risk dengue cluster from the drop-down menu or by tapping its	System displays cluster information in a green card.	System displays cluster information in a green card.

		associated marker.		
2	Medium risk dengue cluster	User selects a medium-risk dengue cluster from the drop-down menu or by tapping its associated marker.	System displays cluster information in a yellow card.	System displays cluster information in a yellow card.
3	High risk dengue cluster	User selects a high-risk dengue cluster from the drop-down menu or by tapping its associated marker.	System displays cluster information in a red card.	System displays cluster information in a red card.

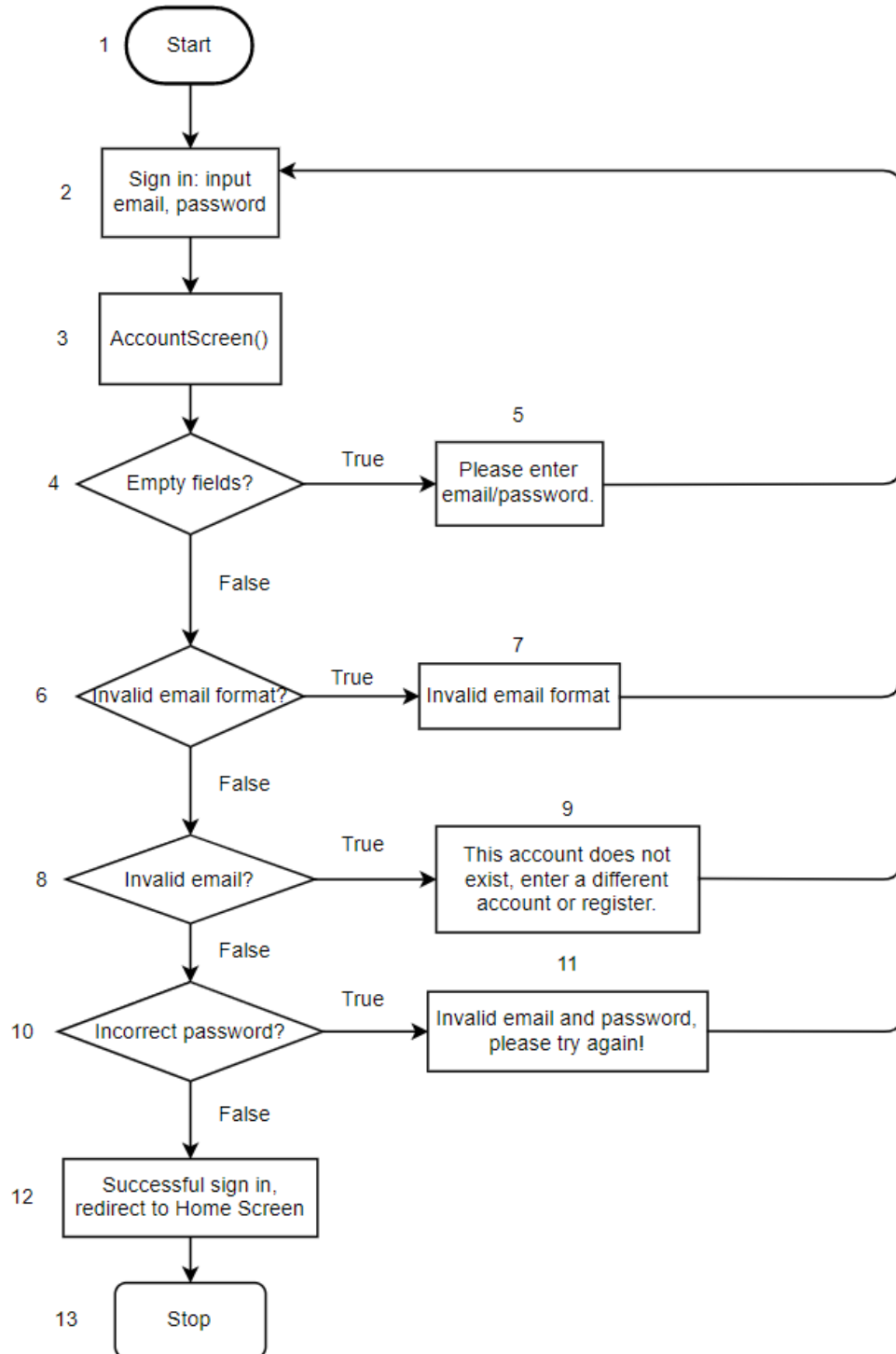
### 1.8. Saved Clusters

Test ID	Test Name	Scenario	Expected Result	Actual Result
1	Add cluster to saved	User saves a cluster into their 'Saved' list.	The system saves the cluster into the user's 'Saved' list.	The system saves the cluster information into the user's 'Saved' list.
2	Remove cluster from saved	User removes a cluster from their 'Saved' list.	The system removes the cluster from the user's 'Saved' list.	The system removes the cluster from the user's 'Saved' list.



## 2. White Box Testing

### 2.1. Sign in



Since all decisions are binary,

Cyclomatic complexity of 'Sign In' = |decisionpoint| + 1

$$= 4 + 1 = 5$$

Therefore, there are a total of 5 basis paths:

BS1	1, 2, 3, 4, 6, 8, 10, 12, 13
BS2	1, 2, 3, 4, 5, 2, 3, 4, 6, 8, 10, 12, 13
BS3	1, 2, 3, 4, 6, 7, 2, 3, 4, 6, 8, 10, 12, 13
BS4	1, 2, 3, 4, 6, 8, 9, 2, 3, 4, 6, 8, 10, 12, 13
BS5	1, 2, 3, 4, 6, 8, 10, 11, 2, 3, 4, 6, 8, 10, 12, 13

The 5 basis paths have the following results:

Basis path	Email	Password	Result
1	denguego@gmail.com	password123	Sign in is successful, the system redirects users to the Home Screen.
2	(Empty)	(Empty)	System prompts the user to enter email and password.
3	abc	password123	System indicates to the user that the email format is invalid.
4	abc@gmail.com	password123	The system will display an error message indicating that the account does not exist.

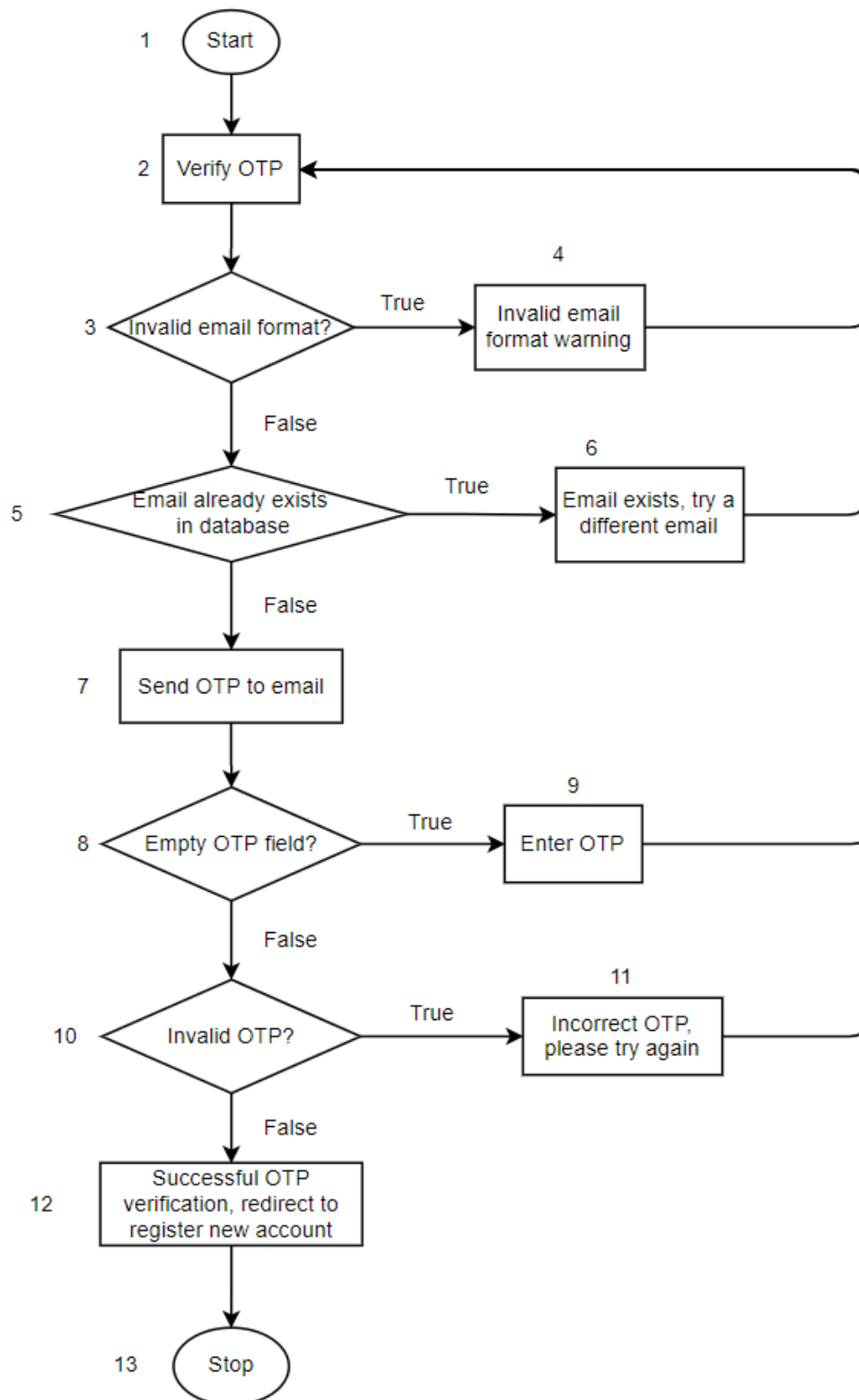
			The system will prompt for the user to log in to a different account or register for a new account.
5	denguego@gmail.com	password456	<p>The system will display an error message indicating invalid email and password.</p> <p>The system will prompt for the user to re-enter their log-in credentials.</p>

Assumptions:

‘[denguego@gmail.com](#)’ is an email found in the database with the corresponding password

‘password123’. ‘[abc@gmail.com](#)’ is an email not found in the database.

## 2.2. Verify OTP



Since all decisions are binary,

Cyclomatic complexity of 'Verify OTP' = |decisionpoint| + 1

$$= 4 + 1 = 5$$

Therefore, there are a total of 5 basis paths:

BS1	1, 2, 3, 5, 7, 8, 10, 12, 13
BS2	1, 2, 3, 4, 2, 3, 5, 7, 8, 10, 12, 13
BS3	1, 2, 3, 5, 6, 2, 3, 5, 7, 8, 10, 12, 13
BS4	1, 2, 3, 5, 7, 8, 9, 2, 3, 5, 7, 8, 10, 12, 13
BS5	1, 2, 3, 5, 7, 8, 10, 11, 2, 3, 5, 7, 8, 10, 12, 13

The 5 basis paths have the following results:

Basis path	Email	OTP	Result
1	newemail@gmail.com	123456	<p>The system will send an OTP to the provided email. The system will inform the user that an email has been sent and prompt the user to check their email.</p> <p>The user will input the OTP received into the 'OTP' field. The system will go to the next screen, where users can fill in their name and password.</p>

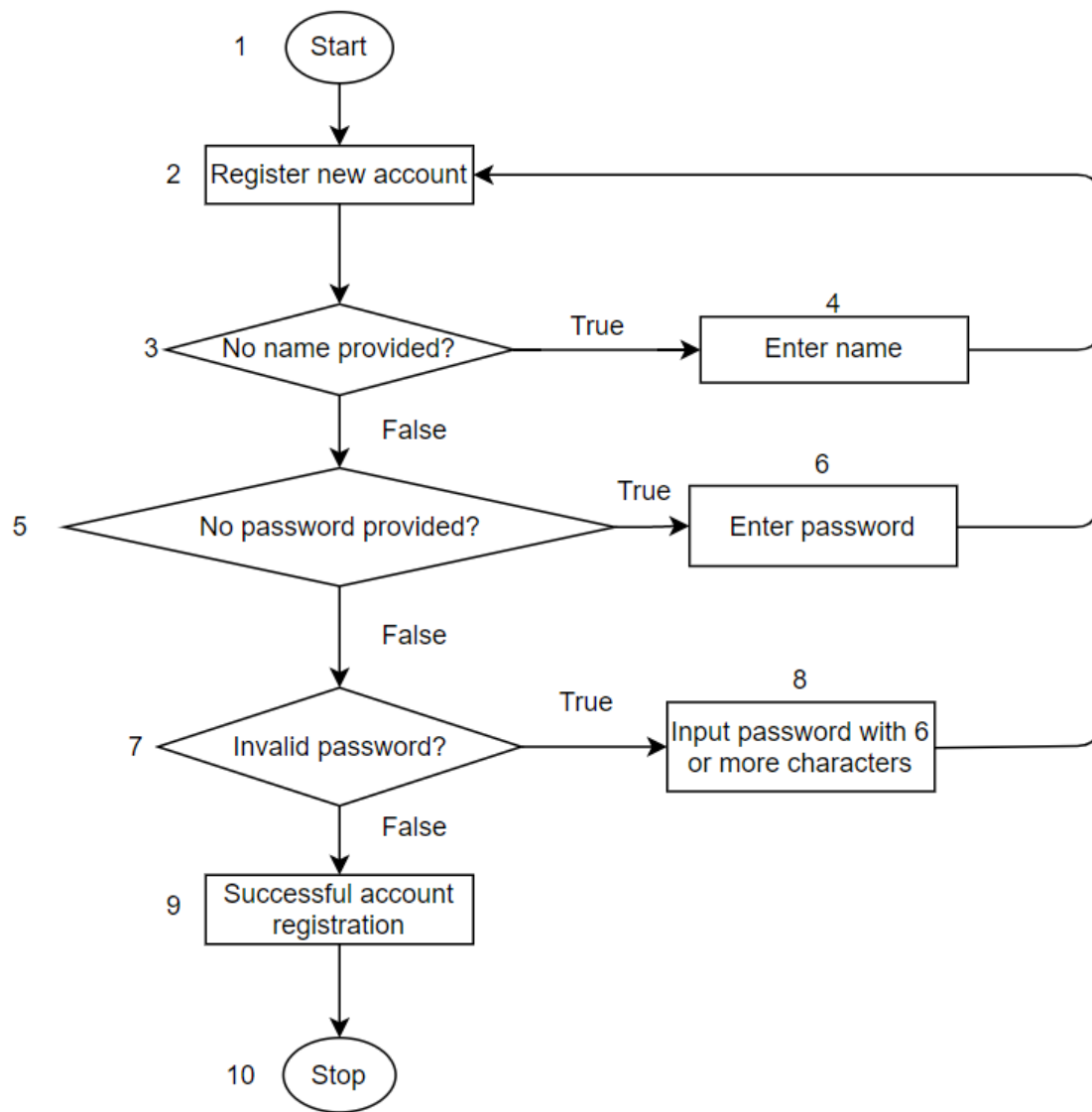
2	<b>newemail</b>	-	The system will display an error message informing the user that the email format is invalid.
3	denguego@gmail.com	-	The system will display an error message informing the user that the email already exists. The system will prompt the user to enter a separate email.
4	newemail@gmail.com	(empty)	The system will display an error message prompting the user to 'enter OTP'.
5	newemail@gmail.com	000000	The system will display an error message informing user that OTP is invalid and prompt for user to re-enter OTP

Assumptions:

'[newemail@gmail.com](mailto:newemail@gmail.com)' is an email that does not exist in the database and can be used to register for a new account.

'[denguego@gmail.com](mailto:denguego@gmail.com)' is an email that belongs to an already-registered user and cannot be used to sign up for a new account. '123456' is the correct OTP.

### 2.3. Register new account



Since all decisions are binary,

Cyclomatic complexity of 'Register new account' = |decisionpoint| + 1

$$= 3 + 1 = 4$$

Therefore, there are a total of 4 basis paths:

BS1	1, 2, 3, 5, 7, 9, 10
-----	----------------------

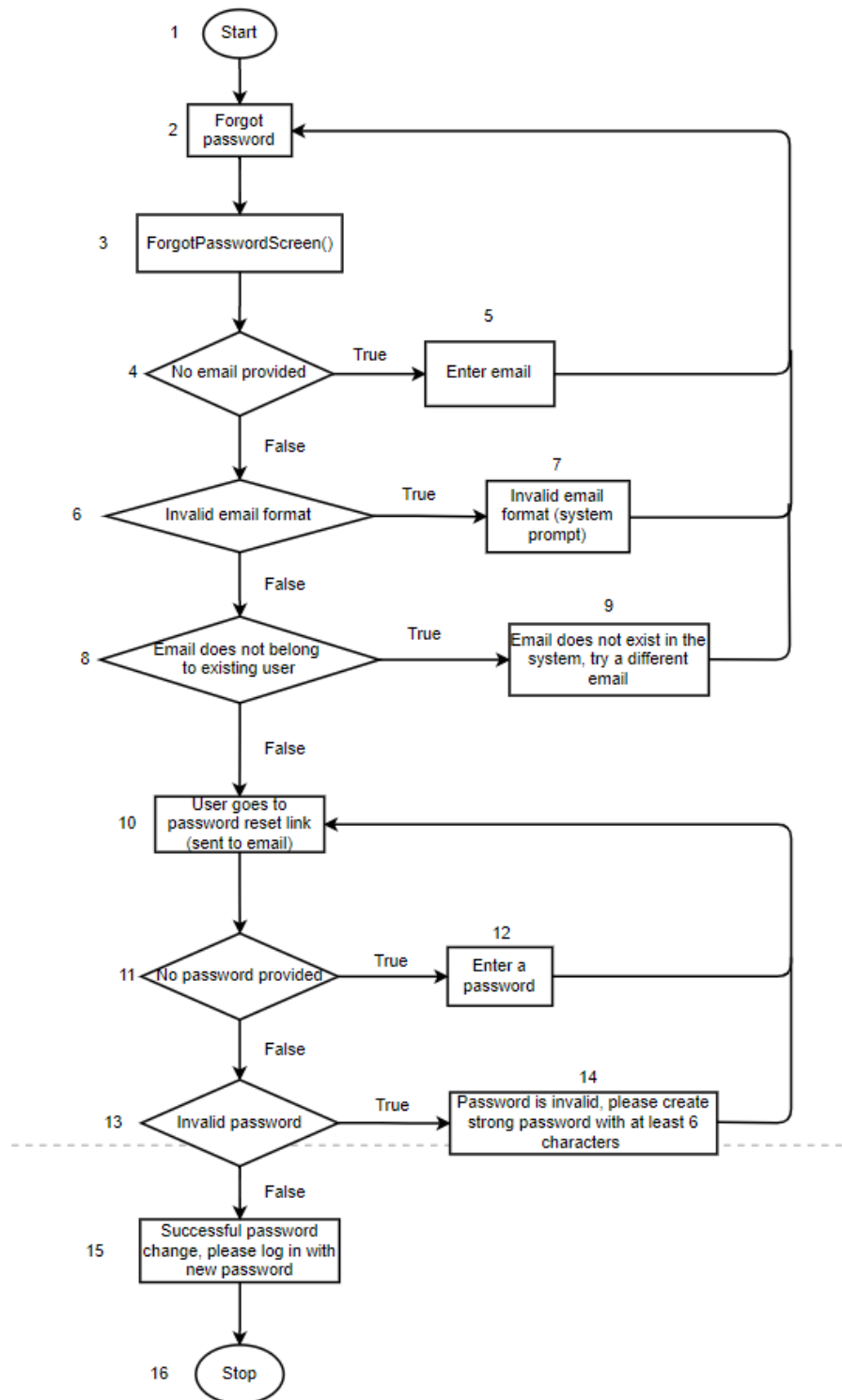
BS2	1, 2, 3, 4, 2, 3, 5, 7, 9, 10
BS3	1, 2, 3, 5, 6, 2, 3, 5, 7, 9, 10
BS4	1, 2, 3, 5, 7, 8, 2, 3, 5, 7, 9, 10

The 4 basis paths have the following results:

Basis path	Name	Password	Result
1	Jessica	password123	The system creates the account and redirects users to the app's Home Screen.
2	(empty)	password123	The system will display an error message prompting the user to enter a name.
3	Jessica	(empty)	The system will display an error message prompting the user to enter a password with 6 or more characters.
4	Jessica	123	The system will display an error message prompting the user to input a password with 6 or more characters



## 2.4. Forgot password



Since all decisions are binary,

Cyclomatic complexity of 'Forgot password' = |decisionpoint| + 1

$$= 5 + 1 = 6$$

Therefore, there are a total of 6 basis paths:

BS1	1, 2, 3, 4, 6, 8, 10, 11, 13, 15, 16
BS2	1, 2, 3, 4, 5, 2, 3, 4, 6, 8, 10, 11, 13, 15, 16
BS3	1, 2, 3, 4, 6, 7, 2, 3, 4, 6, 8, 10, 11, 13, 15, 16
BS4	1, 2, 3, 4, 6, 8, 9, 2, 3, 4, 6, 8, 10, 11, 13, 15, 16
BS5	1, 2, 3, 4, 6, 8, 10, 11, 12, 10, 11, 13, 15, 16
BS6	1, 2, 3, 4, 6, 8, 10, 11, 13, 14, 10, 11, 13, 15, 16

The 6 basis paths have the following results:

Basis path	Email	Password	Result
1	denguego@gmail.com	password123	The system sends an email to the registered user's email containing a link for resetting the password.
2	(empty)	-	The system will display an error message prompting the user to 'Enter email'.

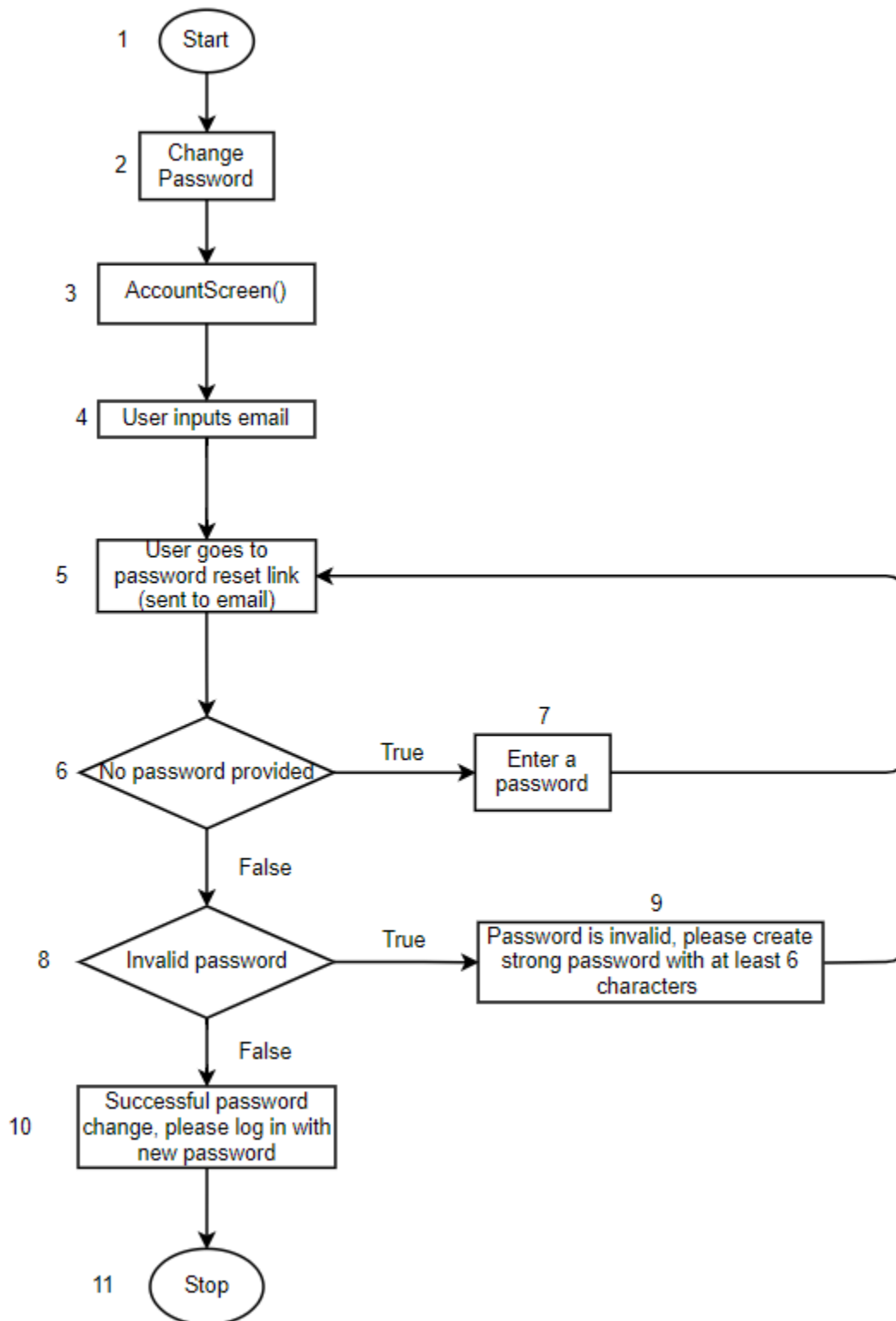
3	denguego	-	The system will display an error message indicating to the user that their input is of an invalid email format.
4	newemail@gmail.com	-	The system indicates to the user that the email does not exist in the system and prompts the user to try a new email.
5	denguego@gmail.com	(empty)	The system displays an error message prompting the user to enter a password.
6	denguego@gmail.com	123	<p>The system indicates to the user that the password they provided is invalid.</p> <p>The system prompts the user to create a strong password containing at least 6 characters.</p>

Assumptions:

'[denguego@gmail.com](mailto:denguego@gmail.com)' is the email of a registered user in the database.

'[newemail@gmail.com](mailto:newemail@gmail.com)' does not correspond to any registered users in the database.

## 2.5. Change password



Since all decisions are binary,

Cyclomatic complexity of 'Change password' = |decisionpoint| + 1

$$= 2 + 1 = 3$$

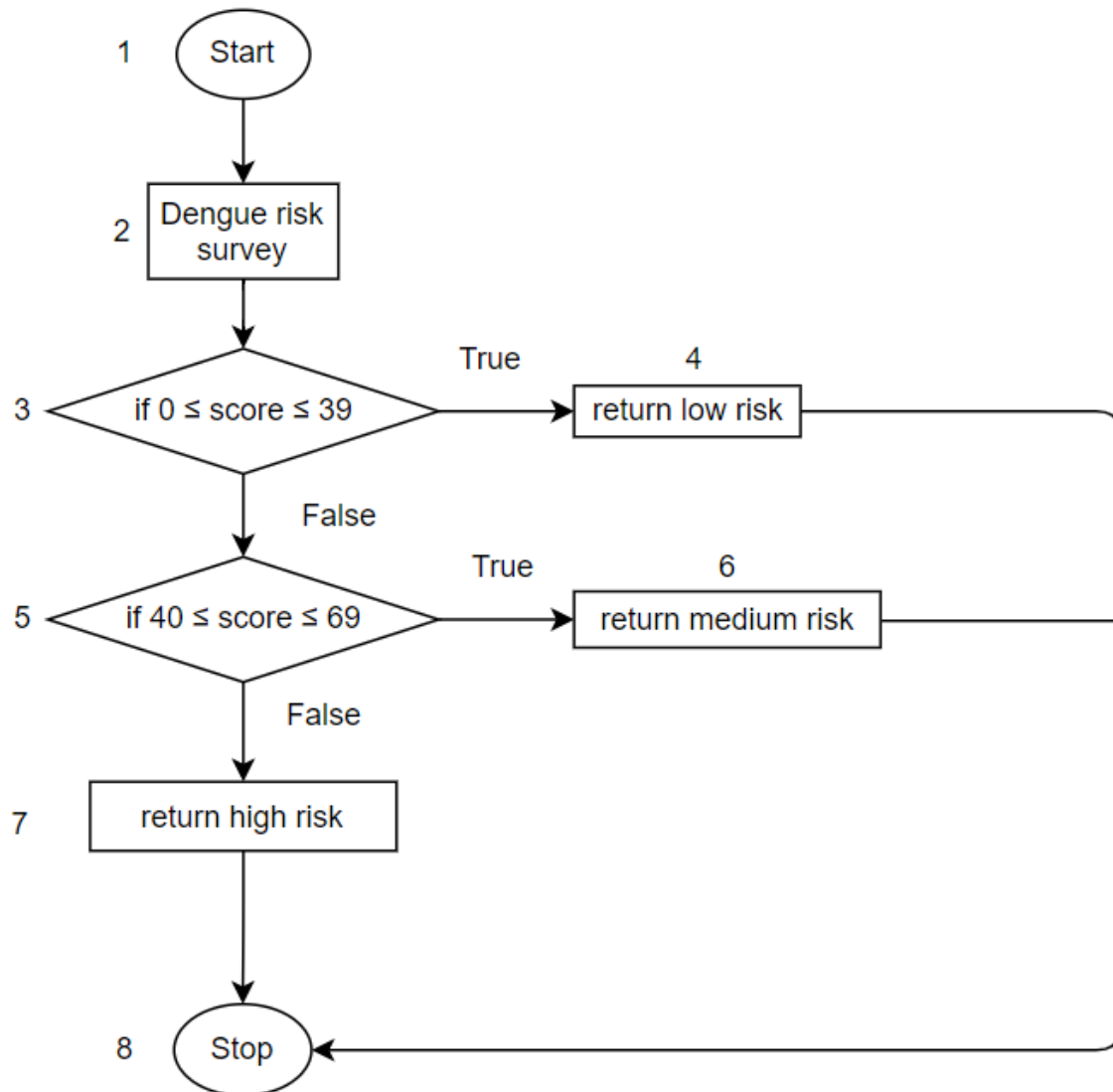
Therefore, there are a total of 3 basis paths:

BS1	1, 2, 3, 4, 5, 6, 8, 10, 11
BS2	1, 2, 3, 4, 5, 6, 7, 5, 6, 8, 10, 11
BS3	1, 2, 3, 4, 5, 6, 8, 9, 5, 6, 8, 10, 11

The 3 basis paths have the following results:

Basis path	Email	Password	Result
1	denguego@gmail.com	password123	The system sends an email to the registered user's email containing a link for resetting the password. The user resets their password.
2	denguego@gmail.com	(empty)	The system displays an error message prompting the user to enter a password.
3	denguego@gmail.com	123	The system indicates to the user that the password they provided is invalid.  The system prompts the user to create a strong password containing at least 6 characters.

## 2.6. Dengue risk survey



Since all decisions are binary,

Cyclomatic complexity of 'Dengue risk survey' = |decisionpoint| + 1

$$= 2 + 1 = 3$$

Therefore, there are a total of 3 basis paths:

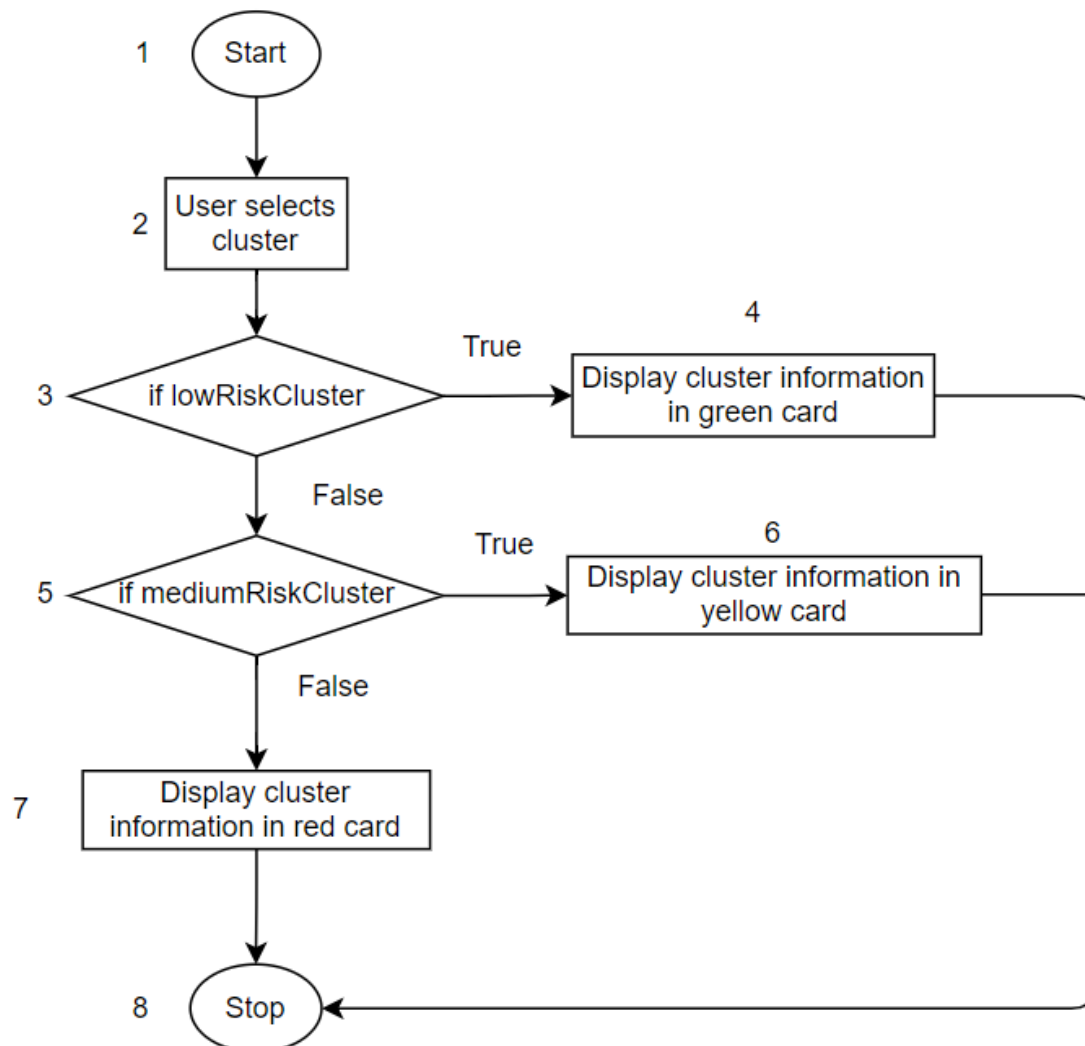
BS1	1, 2, 3, 4, 8
-----	---------------

BS2	1, 2, 3, 5, 6, 8
BS3	1, 2, 3, 5, 7, 8

The 3 basis paths have the following results:

Basis path	Score	Result
1	39	The system shows the user their risk score out of 100 and indicates to them that they are at low risk of contracting dengue.
2	69	The system shows the user their risk score out of 100 and indicates to them that they are at medium risk of contracting dengue.
3	70	The system shows the user their risk score out of 100 and indicates to them that they are at high risk of contracting dengue.

## 2.7. Cluster information



Since all decisions are binary,

Cyclomatic complexity of 'Cluster information' =  $|\text{decisionpoint}| + 1$

$$= 2 + 1 = 3$$

Therefore, there are a total of 3 basis paths:

BS1	1, 2, 3, 4, 8
-----	---------------

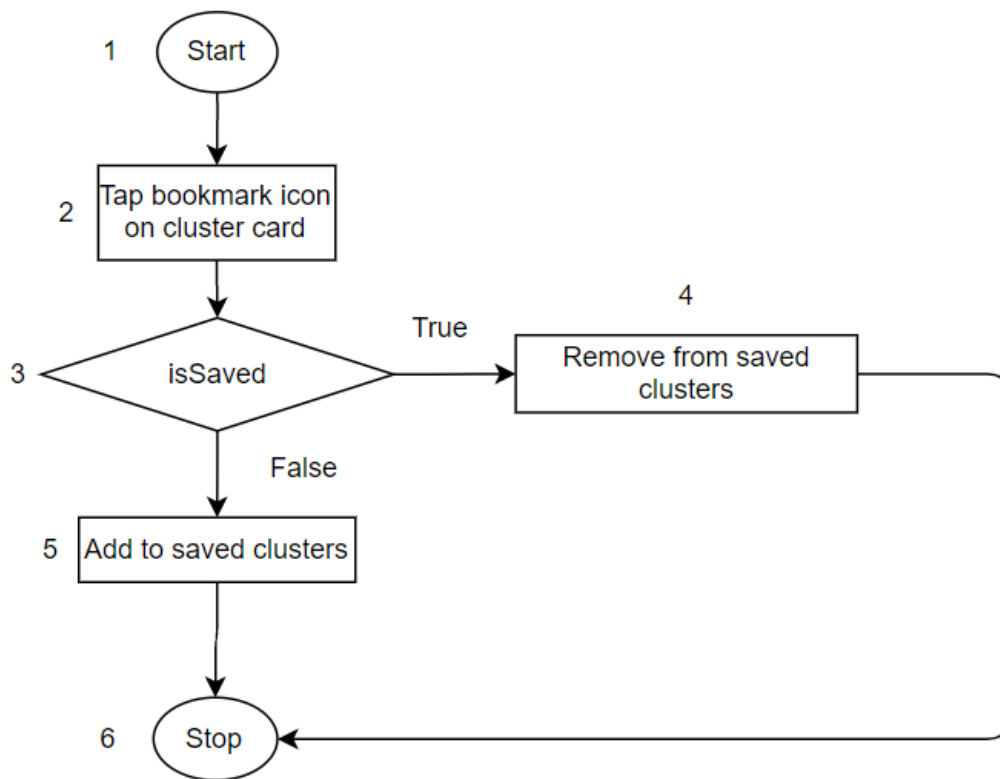


BS2	1, 2, 3, 5, 6, 8
BS3	1, 2, 3, 5, 7, 8

The 3 basis paths have the following results:

Basis path	Cluster	Result
1	Low risk cluster (green marker)	System displays cluster information in a green card.
2	Medium risk cluster (yellow marker)	System displays cluster information in a yellow card.
3	High risk cluster (red marker)	System displays cluster information in a red card.

## 2.8. Saved Clusters



Since all decisions are binary,

Cyclomatic complexity of 'Saved Clusters' = |decisionpoint| + 1

$$= 1 + 1 = 2$$

Therefore, there are a total of 2 basis paths:

BS1	1, 2, 3, 4, 6
BS2	1, 2, 3, 5, 6

The 2 basis paths have the following results:

Basis path	Cluster is already saved	Result
1	True	The system removes the cluster from the user's 'Saved' list.
2	False	The system saves the cluster information into the user's 'Saved' list.