



# SC2006 - Software Engineering

## Lab 1 Deliverables

<b>Lab Group</b>	SCEB
<b>Team</b>	Team 1
<b>Members</b>	AKANKSHA MATHUR (U2323265C)
	HU HAN (U2320036H)
	PHUA NAOMI (U2422699H)
	KARTHIK RAJ S/O MOHAN(U2320917J)
	LI YUJIA (U2320574C)
	WANG KE (U2320276E)

## **Table of Contents**

0. Project introduction.....	3
1. Documentation of functional and non-functional requirements.....	3
1.1 Functional Requirements.....	3
1.2 Non-Functional Requirements.....	4
2. Data dictionary.....	5
3. Initial Use Case Model.....	6
3.1 Use Case diagram.....	6
3.2 Use Case descriptions.....	7
4. UI Mockups.....	19

## **0. Project Introduction**

This is a Public Health and Safety app, a platform for real-time information exchange and alert push on health and safety issues including disease outbreaks, food safety alerts and traffic accidents.

The target users include Singaporean residents who look for real-time information to make informed decisions about their health and safety.

## **1. Documentation of functional and non-functional requirements**

### **1.1 Functional requirement:**

#### **1.1.1 User Authentication and Account Management:**

- Users can log in and create accounts.
- Provide account management options including password recovery and profile updates.

#### **1.1.2 Community Interaction:**

- Enable users to post blogs about safety issues, with the ability to attach relevant tags for categorization.
- Users can interact with posts by liking or disliking them to help verify the legitimacy of the information
- Allow users to report posts as fake news, contributing to the integrity and trustworthiness of the platform.

#### **1.1.3 Information Display and Interaction:**

- Display relevant safety information, sourced from both official APIs and user-submitted reports, in both map and list formats.
- Implement filters and search capabilities that allow users to customize the visibility of community posts and official alerts based on their preferences and location.
- Enable redirection to external apps like Google Maps for extended functionalities.

#### **1.1.4 Real-Time Alerts and Notifications:**

- Provide real-time alerts for critical safety issues such as new disease outbreaks, food safety issues, and major traffic accidents.
- Alerts should be customizable based on user-set preferences and geographic locations.

### **1.1.5 Healthcare Locator and Wait Times:**

- Suggest nearby hospitals and emergency rooms with shorter wait times based on the user's location, highlighting facilities with shorter wait times.
- Offer search functionality on maps to help users find the nearest healthcare facilities with real-time service availability and wait times.

### **1.1.6 Feedback system**

- Allow users to provide feedback on the accuracy and usefulness of the information provided in both posts and alerts.

## **1.2 Non-Functional requirement:**

### **1.2.1 Security**

- User information (e.g passwords) must be encrypted
- Use https for safe connection

### **1.2.2 Reliability**

- Error message displayed after 10 seconds of loading and prompts user to retry
- Updated data that is used in the application
- System must be online and available to use at all times of day, excluding scheduled maintenance
- Critical alerts must be delivered within 10 seconds
- Feedback form for users to bring up any errors that they have encountered to the developers

### **1.2.3 Performance**

- Fast Loading time (10 sec)
- Ability to handle large number of users
- App should be able to locate the users' current location

### **1.2.4 Usability**

- Intuitive Navigation: User friendly UI/UX
- Clear visual design
- Localisation: application can support multiple languages

### **1.2.5 Scalability**

- Must be able to add functions with minimal changes to code

## **2. Data Dictionary**

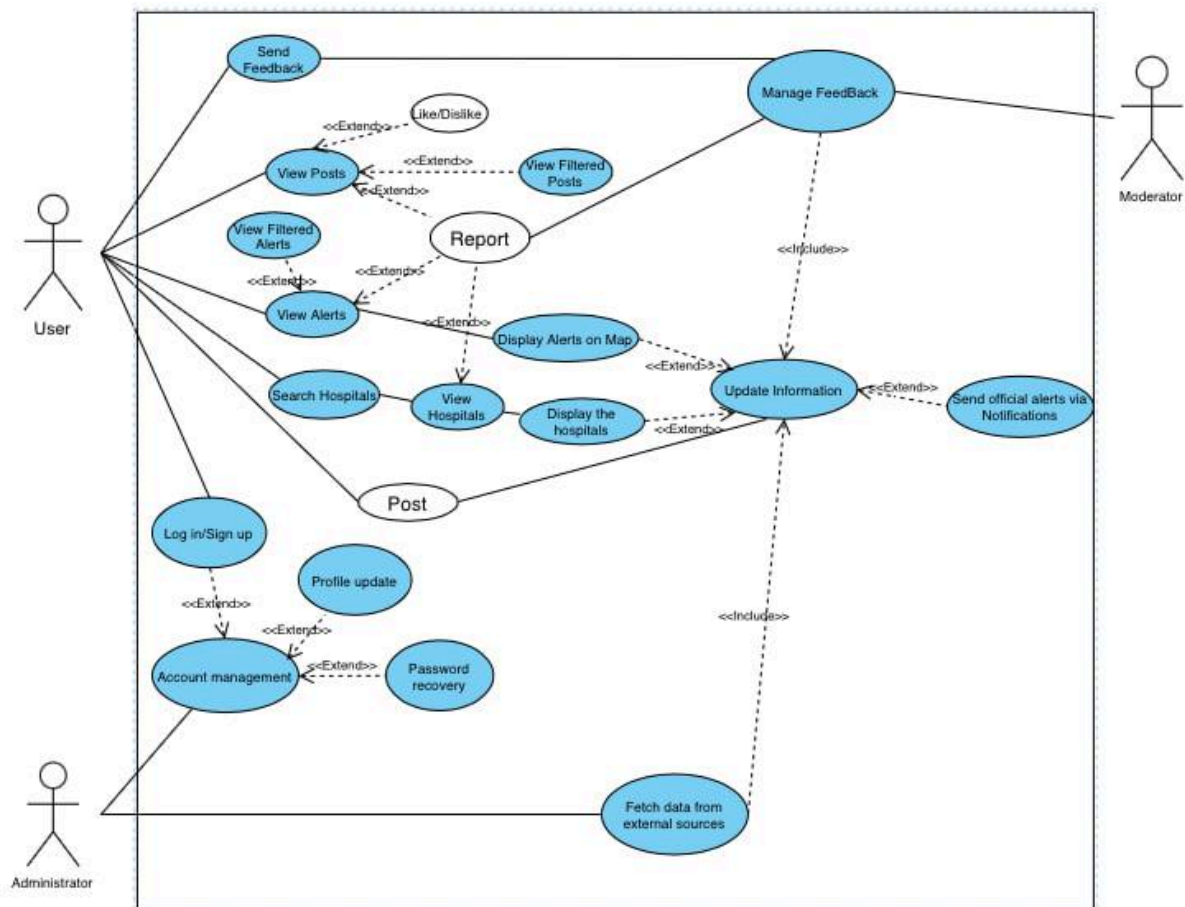
Name	Description
User	User of the application.
Moderator	Respond to reports and verify the reliability and truthfulness of the posts.
Administrator	Manage the services and data, stop violations that happen.
Stakeholder	Officials that provide data APIs and sound alerts.
Data	Data provided by official stakeholders.
Alert	Alerts are generated real-time based on either official announcements or high mention rate of a category or location in the forum. Alert Icons are shown on map and details can be checked upon clicking.
Post	Messages that users upload on the forum, describing the safety condition of a place or announcing an emergency.
Comment	Comment on posts.
Like	An action of upvoting a post.
Dislike	An action of downvoting a post.
Report	An action of reporting a post/alert/hospital information to the moderator.
Category	Different discussion topics in the forum, attached to each post and alert.
Disease Outbreaks	A type of category indicating the outbreak of diseases
Food Safety Issues	A type of category concerning food safety
Traffic Accidents	A type of category concerning traffic accidents on roads.
Location	Indicates the place that a post/alert is mentioning.
Date	Date when a post is uploaded or an alert is generated.
Filter	Filter refers to the search criteria users can set to search for a particular category or location.
Hospital	The hospitals are listed based on the users' search requirements, using data from Google Map and their website.
Map	On the map, users can see the geographical distribution of blogs and alerts, which can be clicked for further information.

Feedback

Users can send feedback to the moderator through the portal.

### 3. Initial Use Case Model

#### 3.1 Case diagram



## 3.2 Case descriptions

### 3.2.1

<b>Use Case Name</b>	Log in/Sign up
<b>Actor</b>	User
<b>Description</b>	Users can log in or sign up to the platform to create and manage their account.
<b>Preconditions</b>	<ol style="list-style-type: none"><li>1. The user device must be connected to WiFi/Cellular Data.</li><li>2. The system currently does not have the user logged in</li><li>3. The user has an existing and verified account. connected to a valid email and password stored in the database to log in.</li><li>4. The hosted database must be online</li></ol>
<b>Postconditions</b>	<ol style="list-style-type: none"><li>1. User is authenticated and can access the platform.</li></ol>
<b>Flow of events</b>	<ol style="list-style-type: none"><li>1. Users input their email and password.</li><li>2. The system verifies the email and password .</li><li>3. If successful, the user is logged in; otherwise, an error message is displayed.</li></ol>

### 3.2.2

<b>Case</b>	Profile Update
<b>Actor</b>	User
<b>Description</b>	Users can update their personal information.
<b>Preconditions</b>	<ol style="list-style-type: none"><li>1. The user must be logged in.</li><li>2. The system must be online and connected to the database</li></ol>
<b>Postconditions</b>	<ol style="list-style-type: none"><li>1. User profile is successfully updated.</li><li>2. The system stores the updated information into the database</li></ol>
<b>Flow of events</b>	<ol style="list-style-type: none"><li>1. User accesses profile settings.</li><li>2. User modifies the necessary details (e.g changing password).</li><li>3. The system saves and updates the profile.</li></ol>

### 3.2.3

<b>Case</b>	Password Recovery
<b>Actor</b>	User
<b>Description</b>	Users can recover their password by providing the registered email.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. User has a registered email.</li> <li>2. The system's email service is online and operational</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Password is reset, and the user can log in.</li> <li>2. The new password is stored in the database</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. User requests a password reset.</li> <li>2. User enters the email associated with the account.</li> <li>3. The system sends a password reset link.</li> <li>4. User resets the password and logs in.</li> </ol>

### 3.2.4

<b>Case</b>	Post
<b>Actor</b>	User
<b>Description</b>	Users can create posts to describe safety issues or emergencies or other relevant things.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. User is logged in.</li> <li>2. The system must be online and connected to the database</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Post is successfully uploaded to the forum.</li> <li>2. Other users can view/comment/like the post</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. User clicks the "Post" button.</li> <li>2. User inputs title, description, and category.</li> <li>3. User attaches tags or categories.</li> <li>4. The system posts the message to the forum.</li> </ol>



### 3.2.5

<b>Case</b>	Edit/Delete Post
<b>Actor</b>	User
<b>Description</b>	Users can modify or delete posts that they created
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. User is logged in.</li> <li>2. The system must be online and connected to the database</li> <li>3. User must be post owner</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. If edited, post is successfully modified and displayed with relevant changed</li> <li>2. if deleted, post is successfully deleted and no longer visible</li> <li>3. The database is updated accordingly</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. User access their post from their profile</li> <li>2. The user makes edits/deletes post</li> <li>3. The system is successfully updated</li> </ol>

### 3.2.6

<b>Case</b>	View Posts
<b>Actor</b>	User
<b>Description</b>	Users can view community posts regarding safety issues.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The user must be logged in.</li> <li>2. The system must be online and connected to the database</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The system successfully displayed the post.</li> <li>2. Users can interact with the post.</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Users access the "View Posts" section.</li> <li>2. System displays a list of posts.</li> </ol>

### 3.2.7

<b>Case</b>	Send Feedback
<b>Actor</b>	User
<b>Description</b>	Users can send feedback regarding posts or alerts to improve the platform.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. User is logged in.</li> <li>2. The system must be online and connected to the database</li> <li>3. The post must be exist and be accessible</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Feedback is sent and stored in the system.</li> <li>2. Moderators may be able to review the content</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. User selects a post or alert to provide feedback on.</li> <li>2. User submits feedback.</li> <li>3. The system processes and stores the feedback</li> </ol>

### 3.2.8

<b>Case</b>	Like/Dislike
<b>Actor</b>	User
<b>Description</b>	Users can like or dislike posts to indicate the usefulness or truthfulness.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. User is logged in and viewing a post.</li> <li>2. The system must be online and connected to the database</li> <li>3. The post must exist and be visible to users</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Post receives a like or dislike vote.</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. User can view the post</li> <li>2. User clicks the "Like" or "Dislike" button.</li> <li>3. The system records the vote.</li> </ol>

### 3.2.9

<b>Case</b>	Report Post
<b>Actor</b>	User
<b>Description</b>	Users can report posts as fake news or inappropriate content.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. User is logged in.</li> <li>2. The system must be online and connected to the database</li> <li>3. The post must exist and be visible to users</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Post is flagged for review by a moderator.</li> <li>2. The system records the report</li> <li>3. If post received multiple reports, post may be hidden</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. User selects a post to report.</li> <li>2. User chooses the reason for reporting or writes other detailed report reasons.</li> <li>3. The system sends the report to the moderator.</li> </ol>

### 3.2.10

<b>Case</b>	View Alerts
<b>Actor</b>	User
<b>Description</b>	Users can view real-time safety alerts on the platform.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. User is logged in.</li> <li>2. The system must be online and connected to the database</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Relevant alerts are displayed based on user settings.</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. User accesses the "View Alerts" section.</li> <li>2. System displays current alerts.</li> </ol>

## 3.2.11

<b>Case</b>	Search Hospitals
<b>Actor</b>	User, Google Maps API
<b>Description</b>	Users can search for nearby hospitals using location-based data.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. User is logged in</li> <li>2. User has location services enabled.</li> <li>3. The system must be online and connected to the database</li> <li>4. The map services must be online</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Hospitals are displayed based on search criteria.</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. User enters location or filter criteria.</li> <li>2. The system fetches hospital data from external sources.</li> <li>3. System displays the hospital list on a map or in a list format.</li> </ol>

## 3.2.12

<b>Case</b>	Display Alerts on Map
<b>Actor</b>	User, Google Maps API
<b>Description</b>	Alerts are displayed on a map for easier location tracking.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. User is logged in.</li> <li>2. User has location services enabled.</li> <li>3. The system must be online and connected to the database</li> <li>4. The map services must be online</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Alerts are marked on the map.</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. User accesses the map.</li> <li>2. System displays alerts with markers on the map.</li> </ol>

### 3.2.13

<b>Case</b>	Send Alerts
<b>Actor</b>	Moderator
<b>Description</b>	Moderators can update alert information based on official sources.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Moderator is logged in and authorized.</li> <li>2. The system must be online and connected to the database</li> <li>3. Alerts must be verified before being sent out</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Alert information is created or updated and stored in the system</li> <li>2. The alerts are sent out to users.</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Moderator receives updated data from external sources.</li> <li>2. Moderator reviews and updates the system with the latest information.</li> <li>3. Alerts are sent to users.</li> </ol>

### 3.2.14

<b>Case</b>	Manage Feedback
<b>Actor</b>	Moderator
<b>Description</b>	Moderators can manage feedback users give.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Moderator is logged in and authorized.</li> <li>2. The system must be online and connected to the database</li> <li>3. User feedback/reports must exist for post</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. User feedback is reviewed</li> <li>2. Information is updated and posts reported are checked.</li> <li>3. Misinforming posts are removed</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>4. Moderator receives feedback and reports.</li> <li>5. Moderator reviews and updates the system with the latest information.</li> <li>6. Fake posts will be deleted.</li> </ol>

## 3.2.15

<b>Case</b>	Encrypt User Data
<b>Actor</b>	System
<b>Description</b>	The system encrypts all users passwords and sensitive information before storing them
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. The user registers or updates personal information</li> <li>2. The system must be online and connected to the database</li> <li>3. The system has encryption mechanisms stored and working</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. User data is securely stored in an encrypted format</li> <li>2. No plain-text personal information is stored in the database</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. User submits sensitive data</li> <li>2. The system applies one-way hashing for passwords</li> <li>3. encrypted data is stored in database</li> </ol>

## 3.2.16

<b>Case</b>	Manage User Accounts
<b>Actor</b>	Administrator
<b>Description</b>	Admins can suspend, delete or restore user accounts due to violations or user request
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin is logged in and authorised</li> <li>2. User accounts exists</li> <li>3. The system must be online and connected to the database</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. User is deleted, suspended or restored</li> <li>2. Changes are stored in the database</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Admin searches for the relevant user</li> <li>2. Admin selects respective action</li> <li>3. System updates user status</li> <li>4. User is notified about the change</li> </ol>

## 3.2.17

<b>Case</b>	Manage User Accounts
<b>Actor</b>	Administrator
<b>Description</b>	Admins can suspend, delete or restore user accounts due to violations or user request
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin is logged in and authorised</li> <li>2. User accounts exists</li> <li>3. The system must be online and connected to the database</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. User is deleted, suspended or restored</li> <li>2. Changes are stored in the database</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Admin searches for the relevant user</li> <li>2. Admin selects respective action</li> <li>3. System updates user status</li> <li>4. User is notified about the change</li> </ol>

## 3.2.18

<b>Case</b>	Manage Moderators
<b>Actor</b>	Administrator
<b>Description</b>	Admins can add, remove or modify moderator permissions
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin is logged in and authorised</li> <li>2. Moderator accounts exists</li> <li>3. The system must be online and connected to the database</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Moderator's roles are modified</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Admin views the moderators profiles</li> <li>2. Admin selects moderator to be modified</li> <li>3. System updates moderator access</li> </ol>

## 3.2.19

<b>Case</b>	Manage Reports and Removes Post
<b>Actor</b>	Administrator
<b>Description</b>	Admins can overview moderation decisions and handle appeals for reported posts/users
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin is logged in and authorised</li> <li>2. Reported post/user exists</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The system updates the post/user modifications</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Admin views reported posts/users</li> <li>2. Admins oversees moderator decisions</li> <li>3. Admin confirms or overrides decisions</li> </ol>

## 3.2.20

<b>Case</b>	Manage Maintenance
<b>Actor</b>	Administrator
<b>Description</b>	Admins controls relevant system configurations (e.g API integration)
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin is logged in and authorised</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The system settings are updated</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Admin access the settings</li> <li>2. Admin modifies the relevant settings</li> <li>3. The system validates input and applies changes</li> <li>4. Users are notified if system is put under maintenance</li> </ol>



### 3.2.21

<b>Case</b>	Sending Notifications and Alerts
<b>Actor</b>	Administrator
<b>Description</b>	Admins can send out system-wide notifications and critical alerts to users
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin is logged in and authorised</li> <li>2. The system must be online and connected to the database</li> <li>3. The alert/notification must be valid</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. The notification is successfully sent out to users</li> <li>2. The systems stores the notification for tracking</li> <li>3. Users receive the notification</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. Admin submits the notification details</li> <li>2. The system validates the notification</li> <li>3. Notification is sent to users via relevant channels</li> <li>4. System stores the notification</li> <li>5. User receives the notification</li> </ol>

### 3.2.22

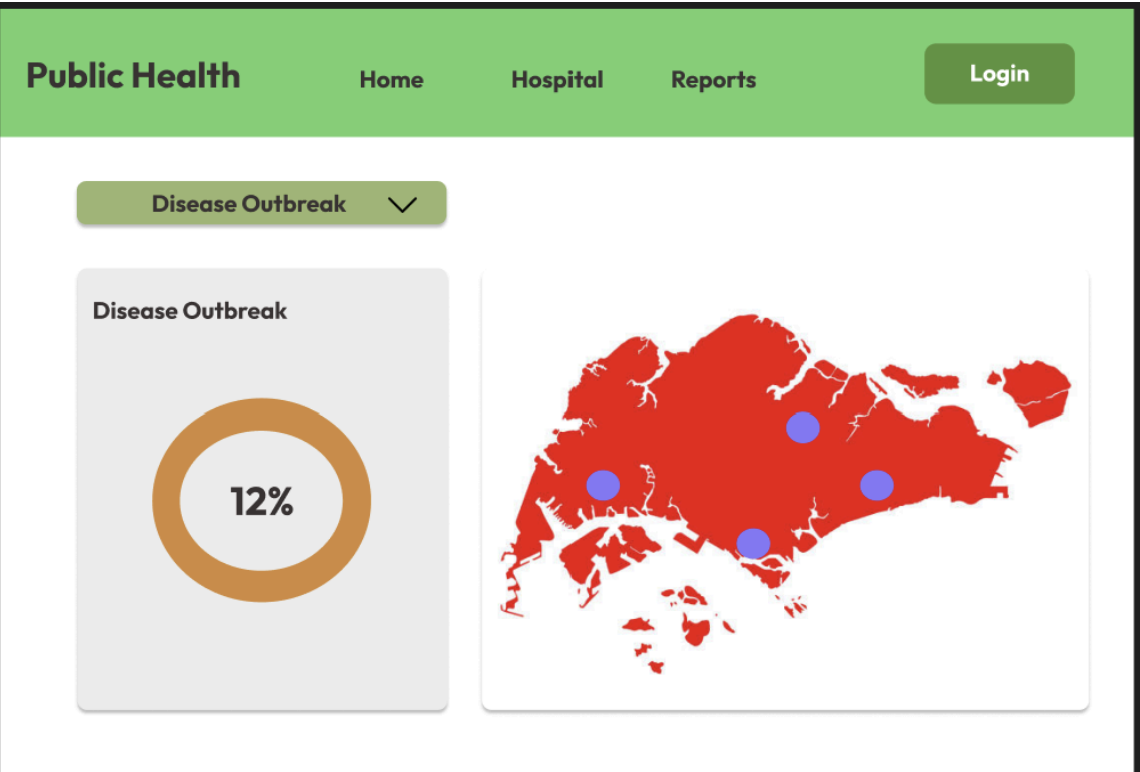
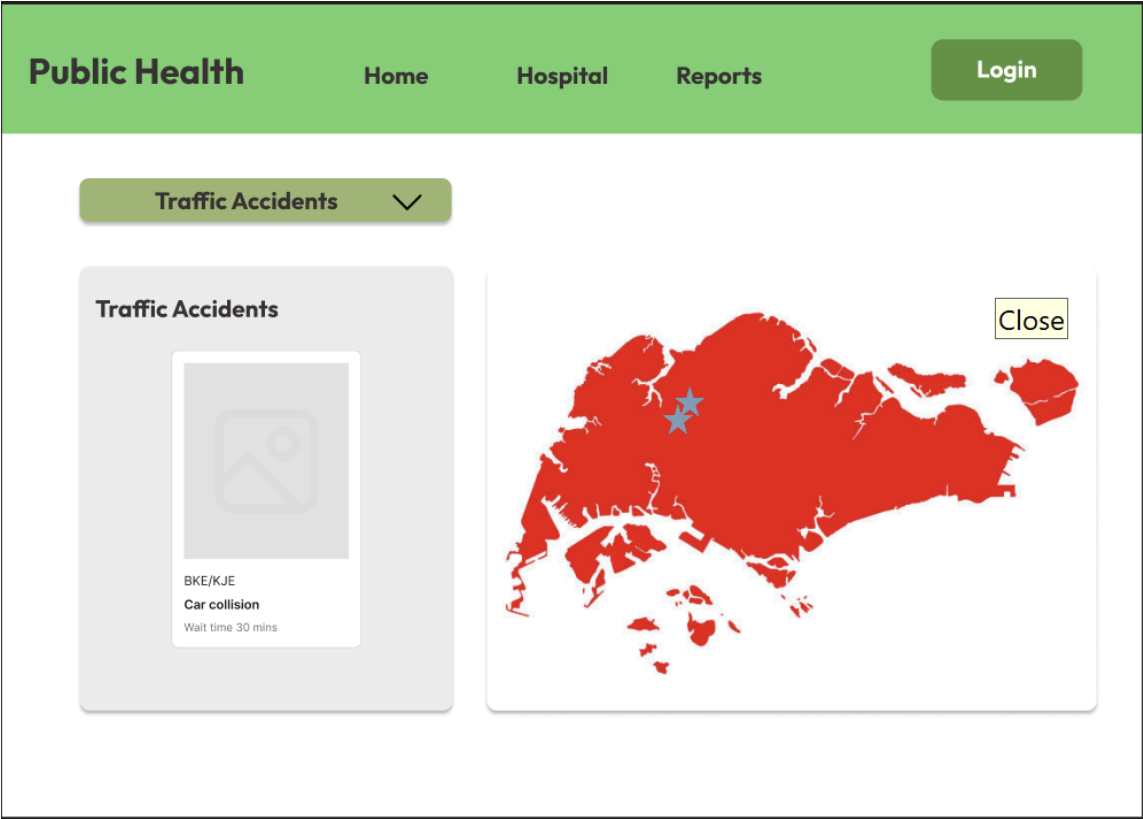
<b>Case</b>	Review and Manage API Integrations
<b>Actor</b>	Administrator
<b>Description</b>	Admins handles the external APIs for alerts, maps and hospitals
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Admin is logged in and authorised</li> <li>2. The system must be online and connected to external APIs</li> <li>3. API keys must be valid and active</li> </ol>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. API configurations are updated and saved in the system</li> <li>2. If new API keys are generated and validated</li> <li>3. The system stores changes for tracking</li> </ol>
<b>Flow of events</b>	<ol style="list-style-type: none"> <li>1. System displays the relevant APIs</li> <li>2. Admin selects an API and modifies the settings</li> </ol>

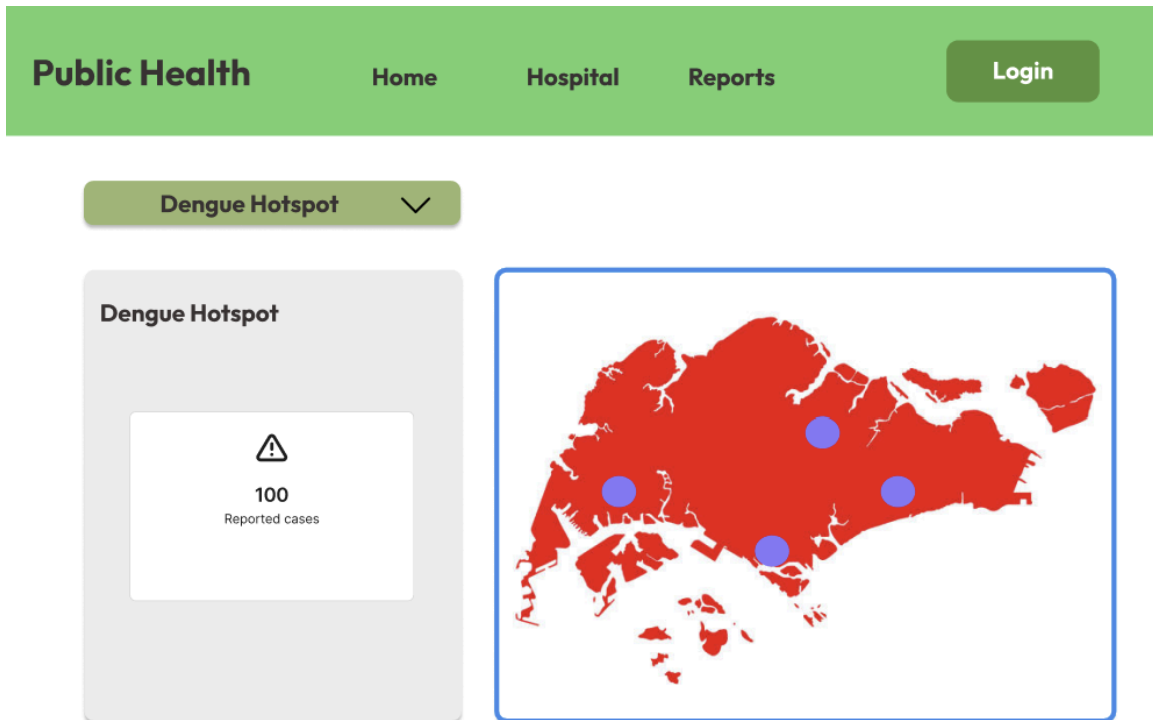
	3. The system saves the changes after validation
--	--

4. UI Mockup

4.1 Homepage

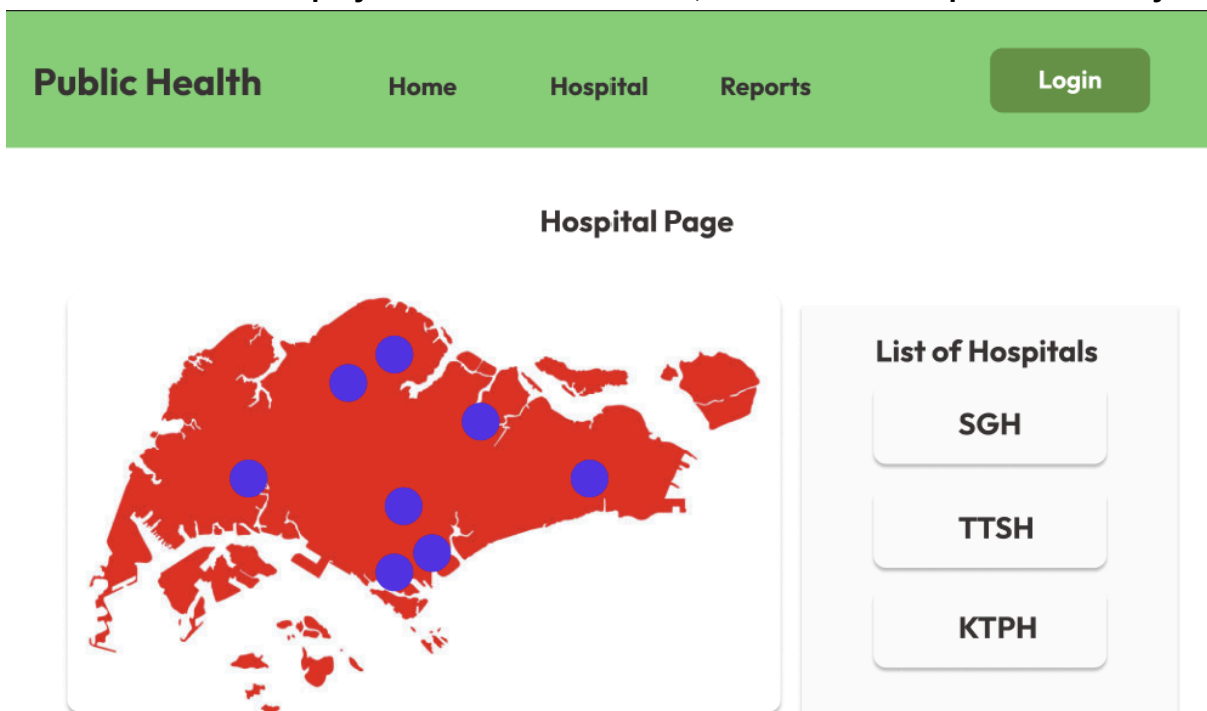
A dashboard to display essential information to the public with an interactive map





#### 4.2 Hospital Page

Users can view the list of hospitals available in Singapore. Once clicked relevant information will be displayed such as the location, wait time and car park availability



Public Health

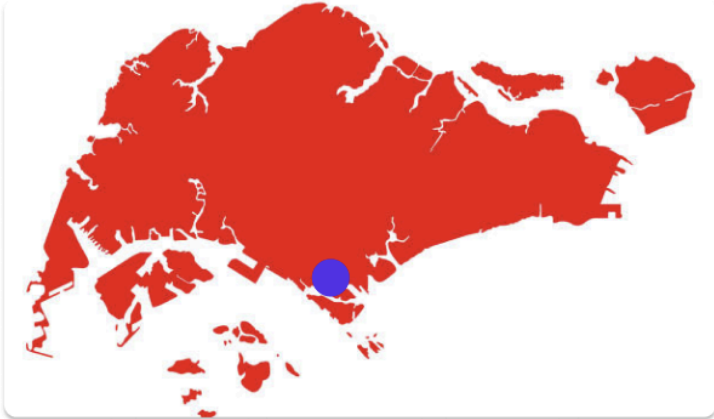
Home

Hospital

Reports

Login

SINGAPORE GENERAL HOSPITAL - SGH



Address:

31 Third Hospital Ave, Singapore 168753

Live wait time:

30:00 mm:ss

Carpark Availability:

112

#### 4.3 Sign Up Page

Users can sign up here when they first use the app

Public Health

Home

Hospital

Reports

Sign up

Name

Password

Re-enter Password

Mobile No

Email

Sign Up

#### 4.4 Login Page

Users can choose to login to view their personal information and post latest medical related news

**Public Health**HomeHospitalReports

### Login

Name

Password

LOGIN

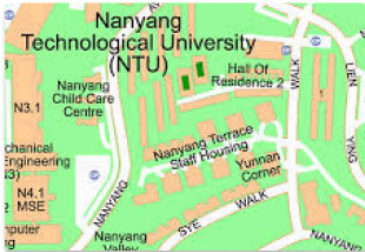
Not registered? Sign up [here](#)

#### 4.5 Profile - Contact Help page

User can view this page when facing technical issues

**Public Health**HomeHospitalReports

Welcome, Name



Office:

NANYANG TECHNOLOGICAL UNIVERSITY,SINGAPORE  
50 Nanyang Ave, Singapore 639798

Phone:

+65 25369842

Profile

Feedback

Contact Help

Logout

Need help with your account?Contact the Admin here: Admin123@gmail.com

#### 4.6 Profile - Feedback page

User can provide feedback and submit their queries

The screenshot shows the 'Public Health' app interface. The top navigation bar is green with white text for 'Public Health', 'Home', 'Hospital', and 'Reports'. On the right is a user profile icon. Below the navigation bar, the text 'Welcome, Name' is displayed. To the right of the welcome message is a dropdown menu with options: 'Profile', 'Feedback' (highlighted in light green), 'Contact Help', and 'Logout'. Below the welcome message, the text 'Submit Feedback' is shown. The main form is a green rounded rectangle containing a white dropdown menu labeled 'Choose topic' with a downward arrow, and a white text input field labeled 'Description'. Below the form is a green 'Submit' button.

#### 4.7 User profile page

User can view their personal information and post a report which will be updated in the forum

The screenshot shows the 'Public Health' app interface. The top navigation bar is green with white text for 'Public Health', 'Home', 'Hospital', and 'Reports'. On the right is a user profile icon. Below the navigation bar, the text 'Welcome, Name' is displayed. To the right of the welcome message is a dropdown menu with options: 'Profile' (highlighted in light green), 'Feedback', 'Contact Help', and 'Logout'. Below the welcome message, there are two green buttons: 'Mobile Number' and 'Email'. Below these buttons, the text 'Post Report' is shown. The main form is a green rounded rectangle containing a white text input field labeled 'Title' and a white text input field labeled 'Description'. Below the form is a green 'Post' button.

#### 4.8 Report forum page

Users can view reports made by other users as well as interact with posts by liking or disliking them.

