

---

# **Software Requirements Specification**

**for**  
**Safe, Smart, Strong**

**Version 1.0 approved**

**Prepared by:**  
**AKANKSHA MATHUR (U2323265C)**  
**HU HAN (U2320036H)**  
**PHUA NAOMI (U2422699H)**  
**KARTHIK RAJ S/O MOHAN(U2320917J)**  
**LI YUJIA (U2320574C)**  
**WANG KE (U2320276E)**

**Team 1**

**12 April 2025**

## Table of Contents

<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	2
1.5 References	2
<b>2. Overall Description</b>	<b>2</b>
2.1 Product Perspective	2
2.2 Product Functions	3
2.3 User Classes and Characteristics	5
2.4 Operating Environment	6
2.5 Design and Implementation Constraints	6
2.6 User Documentation	6
2.7 Assumptions and Dependencies	7
<b>3. External Interface Requirements</b>	<b>7</b>
3.1 User Interfaces	7
3.2 Hardware Interfaces	15
3.3 Software Interfaces	15
3.4 Communications Interfaces	16
<b>4. System Features</b>	<b>17</b>
4.1 User Authentication	17
4.2 Profile Update	18
4.3 Password Recovery	18
4.4 Create Post	19
4.5 Edit/Delete Post	19
4.6 View Posts	20
4.7 Send Feedback	20
4.8 Like Post	21
4.9 Report Post	21
4.10 View Alerts	22
4.11 Search Hospitals	22
4.12 Display Alerts on Map	23
4.13 Send Alerts	23
4.14 Manage Feedback	24
4.15 Encrypt User Data	24
4.16 Manage User Accounts	25
4.17 Manage Moderators	25
4.18 Manage Reports and Post Removals	26

*Software Requirements Specification for Safe, Smart, Strong*

4.19 Manage Maintenance	26
4.20 Sending Notifications and Alerts	27
4.21 Review and Manage API Integrations	27
<b>5. Other Nonfunctional Requirements</b>	<b>28</b>
5.1 Performance Requirements	28
5.2 Safety Requirements	28
5.3 Security Requirements	28
5.4 Software Quality Attributes	28
5.5 Business Rules	29
<b>6. Other Requirements</b>	<b>30</b>
<b>Appendix A: Glossary</b>	<b>31</b>
<b>Appendix B: Analysis Models</b>	<b>32</b>
<b>Appendix C: To Be Determined List</b>	<b>34</b>

# 1. Introduction

## 1.1 Purpose

This document details the software requirements and specifications for **Safe, Smart, Strong**, a web application designed to help Singapore residents in accessing health and safety resources, including browsing health-focused news articles, searching for nearby Healthcare facilities, urgent health alerts, and creating or viewing community blog posts. This document will detail the frontend user interactions, integration with external APIs, and backend support. It is intended for both general public users and contributors such as healthcare officials and community volunteers.

## 1.2 Document Conventions

Text Format

Font: Times New Roman

Font Size: 12 for Body, 14 for Sub-Headings, 18 for Headings

Line Spacing: 1.15

## 1.3 Intended Audience and Reading Suggestions

This document is intended for all the stakeholders involved in the development, testing, design, and deployment of **Safe, Smart, Strong**. Its purpose is to provide a comprehensive understanding of the systems' functionality, design, and implementation scope. The following is a summary of each part of the document:

**Part 1 (Introduction):** Provides an overview of the **Safe, Smart, Strong** project, outlining the project scope, intended audience, key objectives, and the outline of the document structure.

**Part 2 (Overall Description):** This section details the **Safe, Smart, Strong** application. It described the general functionality of the project, including its main features, user types, performance/design constraints, and system environment. It will also introduce relevant class relationships and diagrams

**Part 3 (External Interface Requirements):** This section documents the user interface design, including key components of the **Safe, Smart, Strong**. It will describe how the system interacts with external components such as user interfaces, APIs, hardware constraints, and communication protocols for external services.

**Part 4 (System Features):** This section highlights the functional requirements of the application in detail. Each system detail - such as news browsing, searching for nearby healthcare facilities, or creating blog posts - is described with corresponding expected behaviour and system responses. Readers can view this section to see what the product looks like.

**Part 5 (Other Nonfunctional Requirements):** This section outlines the nonfunctional requirements of the application - such as the privacy and security requirements - to ensure that the **Safe, Smart, Strong** is usable, reliable, and secure for both users and developers.

**Part 6 (Other Requirements):** This section defines other supporting documentation - such as a data dictionary, test case summaries, testing and expected outputs, and other relevant information.

## 1.4 Product Scope

The **Safe, Smart, Strong** is a web application produced by Team 1 which aims to be a digital platform for Singaporean residents to access various essential health and safety resources and information. The system incorporates features which allow users to browse health related news, search for nearby healthcare facilities, receive urgent health alerts, and engage with the community through blogs. Our platform aims to improve public awareness and access to health resources through a responsive and user friendly interface, real-time content delivery, and streamlined interaction with external APIs and backend services.

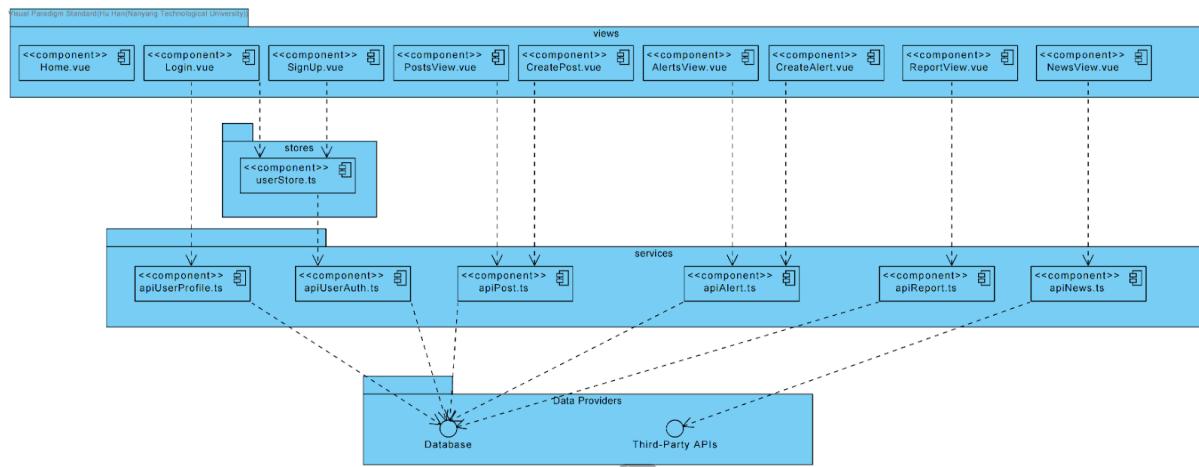
## 1.5 References

*<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>*

# 2. Overall Description

## 2.1 Product Perspective

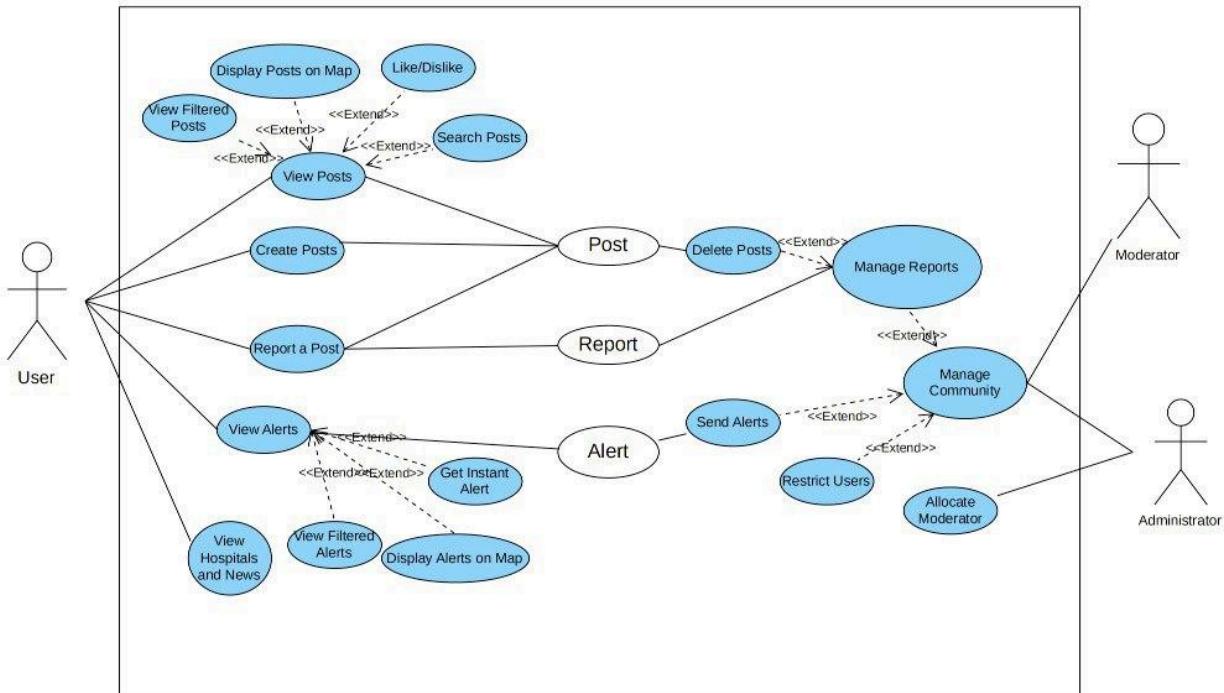
The **Safe, Smart, Strong** is a new, self contained, web application developed as a standalone system. It is not a replacement or upgrade of an existing software solution. Instead, it was designed from scratch to serve as a centralized digital health and safety platform for residents in Singapore. It integrates data from external APIs to deliver health-focused content and location based information, and provides a platform for users to interact with this information. The system architecture diagram (shown below) illustrates the connections and external interfaces.



## 2.2 Product Functions

### 2.2.1 Use Case Diagram

The diagram below presents the key features of the **Safe, Smart, Strong**. There are 3 main users - Admin, Moderator, and User. Each type of user interacts with the **Safe, Smart, Strong** based on their role-specific functions.



## 2.2.2 Main Product Functions

The **Safe, Smart, Strong** provides a set of essential features designed to assist residents in accessing relevant, timely, location-based information.

### User Functions:

1. Authentication
  - a. Log in/Sign up
  - b. Password Recovery
  - c. Profile Updates
2. Community Interactions
  - a. Create/Edit/Delete/View Blog Posts
  - b. Like and Comment on Posts
  - c. View Safety Alerts
3. Health and Safety Tools
  - a. Search for nearby Healthcare facilities
  - b. Views alerts on interactive map
  - c. Access real-time health news

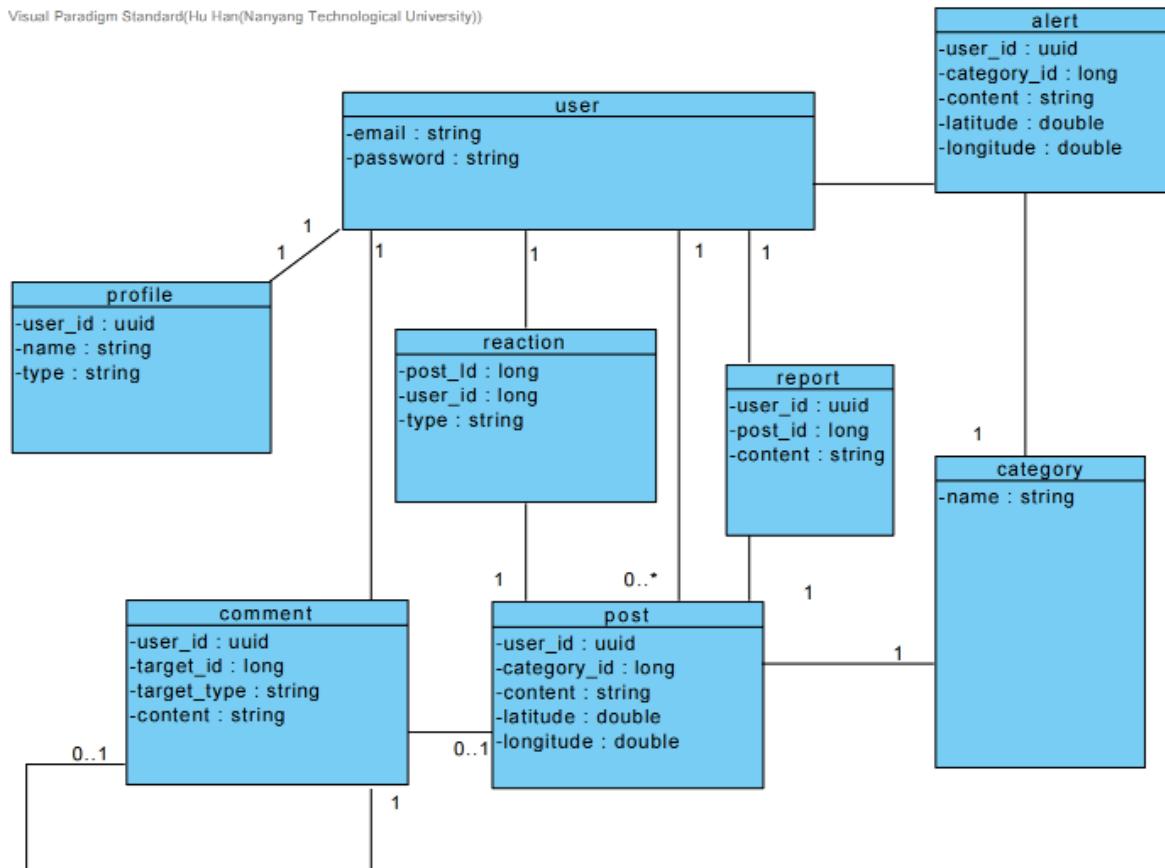
### Moderator Functions:

1. Review and Respond to user posts
2. Moderate flagged content
3. Push alerts to users
4. Manage user feedback

### Admin Functions:

1. Manage user accounts
2. Add/Remove/Modify Moderator permissions
3. Oversee moderation actions
4. Configure system integrations
5. Send system-wide alerts and notifications
6. Handle API key management and platform maintenance

### 2.2.3 Class Diagram



## 2.3 User Classes and Characteristics

Users of the **Safe, Smart, Strong** can be categorised as:

- General Users:** Singapore Residents using the platform to access health and safety resources. They require simple, intuitive interfaces and prioritise fast access real-time content.
- Moderators:** Users responsible for maintaining the integrity and accuracy of user-generated content. They require technical skills in content moderation, and community policies.
- Administrators:** Team members responsible for overseeing backend system operations and overall platform governance. They require high technical proficiency and should be trusted with sensitive operations and system security.

All 3 users of the platform are equally important as they form an essential pipeline that ensures the platform remains functional and reliable. The general users provide the core activity and engagement, moderators review this activity, and admins ensure the stability and security of the system.

## 2.4 Operating Environment

The application is web based, optimised for both desktop and mobile use, designed to operate across a wide range of devices and platforms. It will run on the latest versions of Chrome, FireFox, Safari, and Edge. It will be hosted on cloud services to ensure high availability and scalability and integrate external APIs (Google Maps API, NewsAPI, data.gov.sg). Data storage and authentication are managed with Supabase, which provides a secure PostgreSQL database.

## 2.5 Design and Implementation Constraints

1. **Technology Stack Constraints:** The frontend must be developed using Vue.js with Typescript, and Tailwind CSS. The backend and database are handled using Supabase which limits backend logic to Supabase functions or client-side integration
2. **API Dependencies:** The system relies on free third-party APIs, which may reduce feasibility of the application. Any limitations in rate, availability, or format of the APIs will directly impact the application.
3. **Hosting and Deployment:** The application must be deployed on a cloud-based platform which limits deployment formats, build processes, and environment variable handling.
4. **Security and Privacy Constraints:** All sensitive data must be hashed and stored securely. The system must follow data protection regulations and ensure secure authentication and role-based authorisation mechanisms

## 2.6 User Documentation

The platform will provide detailed documentation of how all the users interact with the system to support ease of use, onboarding, and future maintenance. The documentation components include:

1. **README Files**
  - 1.1 **Main Repository:** A master README.md will present an overview of the **Safe, Smart, Strong** architecture, setup instructions, deployment steps, and links to the frontend/backend folders. It will describe the tech stack used (Vue.js, Typescript, Supabase, Tailwind CSS), and versioning details.
  - 1.2 **Frontend Repository:** The frontend directory will have a README.md outlining the project setup and environment configuration, explanation of the folder structure, and guidelines to using the tech stack
  - 1.3 **Backend Repository:** The backend directory will have a README.md with instructions on setting up and connecting to Supabase, the environment variable setup, usage of Supabase features, and access control.
2. **Code Comments:** All source codes for frontend and backend will include clear comments on components, complex logic, functions, classes, and methods.
3. **API Documentation:** While Supabase provides Backend-as-a-service, the application will provide documentation of all interactions with external and internal APIs.

## 2.7 Assumptions and Dependencies

### Assumptions:

1. There will be stable internet connection to access the platform
2. Users will allow access to location services
3. External APIs will remain stable, accessible, and consistently provide data to users.

### Dependencies:

1. **Google Maps API:** For geolocation services, map depiction, and location-based search functions.
2. **NewsAPI:** To retrieve and display health-focused relevant news articles.
3. **Supabase:** Provides BaaS for database management, authentication, storage, and real-time updates.
4. **data.gov.sg Environmental API:** Provides relevant weather data (weather forecast, temperature, humidity, wind data, and rainfall information) to support real-time public safety alerts.

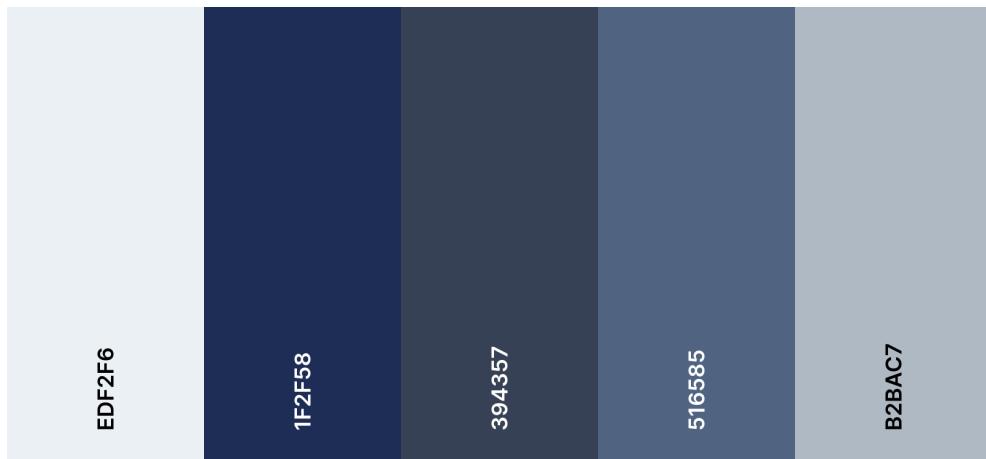
## 3. External Interface Requirements

### 3.1 User Interfaces

The **Safe, Smart, Strong** provides an intuitive and responsive user experience across desktop and mobile platforms. The design uses modern web usability standards with a clean layout, accessibility concerns, and user-friendly navigation.

#### 3.1.1 Style Guides

##### 3.1.1.1 Theme Colour Scheme



### **3.1.2 General User Interface**

#### **4.1 Sign In Page**

The screenshot shows the SG Health Portal's sign-in interface. At the top, there is a dark blue header bar with the portal's name "SG Health Portal" on the left and a "Sign In" button on the right. Below the header, the main content area features a light gray background with a central white rectangular form. The form has a title "Sign in to your account" at the top. It contains two input fields: "Email address" and "Password". Below these fields is a dark blue "Sign in" button. Underneath the button, there are two links: "Don't have an account? Sign up" and "Forgot password?".

#### **4.2 Sign Up Page**

The screenshot shows the SG Health Portal's sign-up interface. Similar to the sign-in page, it has a dark blue header bar with the portal's name "SG Health Portal" on the left and a "Sign In" button on the right. The main content area is a light gray page with a central white rectangular form. The form has a title "Create your account" at the top. It contains three input fields: "Email address", "Password", and "Confirm Password". Below these fields is a dark blue "Sign up" button. Underneath the button, there is a link: "Already have an account? Sign in".

## 4.3 Home Page

**SG Health Portal** Home Posts News Sign In

**Jalan Bahar**  
Monday, 14 April 2025  
12:25 am  
Data from data.gov.sg

Humidity 89% Wind 1.6 m/s WNW Rainfall 0 mm

Updated: 14/04/2025, 12:13:00 am

**Data Layers**

**Post Locations**

- Hotspots

**Active Layer: Post Locations**

- Post Location
- Hotspot Area (K-Means Clustering)

**Hotspot Coordinates**

- 1.34661, 103.83267
- 1.32879, 103.69433
- 1.37762, 103.95435
- 1.32879, 103.69433
- 1.36012, 103.84689

**Health Tips & Resources**

**COVID-19 Updates**  
Stay informed about the latest COVID-19 measures, vaccination programs, and health advisories.  
[Learn more →](#)

**Wellness Programs**  
Discover community wellness programs, fitness activities, and mental health resources.  
[Find programs →](#)

**Health Screening**  
Learn about recommended health screenings and find affordable screening programs.  
[Check eligibility →](#)

Data provided by Singapore Government APIs, Google Maps, and simulated for demonstration purposes.

#### **4.4 Profile Page**

### User Profile

[Edit Profile](#)

---

#### Basic Information

Full Name  
TestAccount

User Type:  
NORMAL

#### **4.5 Submit Feedback Page**

### Submit Feedback

---

Feedback Type  
General Feedback

Your Message  
Please share your thoughts...

[Submit Feedback](#)

---

Change Password

## 4.6 Healthcare Facilities

**SG Health Portal**

- Home
- Healthcare Facilities
- Health Blog
- Posts
- Alerts
- [Sign In](#)

### Healthcare Facilities

All Facilities
Hospitals
Polyclinics

- Singapore General Hospital  
Hospital  
Outram Road, Singapore 169608
- Tan Tock Seng Hospital  
Hospital  
11 Jalan Tan Tock Seng, Singapore 308433
- KK Women's and Children's Hospital  
Hospital  
100 Bukit Timah Road, Singapore 229899
- Clementi Polyclinic  
Polyclinic  
Blk 451 Clementi Ave 3, Singapore 120451

**Legend**

- Hospital
- Polyclinic

The map displays the locations of several healthcare facilities in Singapore, including Singapore General Hospital, Tan Tock Seng Hospital, KK Women's and Children's Hospital, and Clementi Polyclinic. The map covers areas such as Outram, Tanjong Katong, and Clementi.

- Tan Tock Seng Hospital  
Hospital  
11 Jalan Tan Tock Seng, Singapore 308433
- KK Women's and Children's Hospital  
Hospital  
100 Bukit Timah Road, Singapore 229899
- Clementi Polyclinic  
Polyclinic  
Blk 451 Clementi Ave 3, Singapore 120451

**Legend**

- Hospital
- Polyclinic

A detailed map of the Singapore General Hospital area, showing its proximity to other landmarks like the Singapore Zoo, Marina Bay, and the CBD. A callout box highlights the hospital's specialties: General Medicine, Cardiology, and Oncology.

**Singapore General Hospital**

Hospital

**Contact Information**

Outram Road, Singapore 169608  
+65 6222 3322

**Specialties**

- General Medicine
- Cardiology
- Oncology

**Services**

Emergency Services ✓

Lab Services ✓

Outpatient Care ✓

Pharmacy ✓

**Operating Hours**

Weekdays 8:30 AM - 5:30 PM	Weekends 24 hours
-------------------------------	----------------------

[Get Directions](#)

## 4.7 Health Blog

SG Health Portal    Home    Healthcare Facilities    Health Blog    Posts    Alerts    Sign In

### Latest Health News



Ars Technica  
**Trump White House budget proposal eviscerates science funding at NASA - Ars Technica**  
"This would decimate American leadership in space..."

April 11, 2025    [Read More →](#)



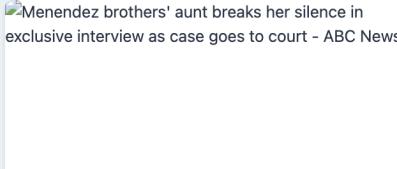
CBS Sports  
**2025 NFL mock draft: Giants trade down, Saints move into top five for QB, Cowboys add big WR - CBS Sports**  
Cody Benjamin forecasts the first round of the draft

April 11, 2025    [Read More →](#)



NBCSports.com  
**Joe Flacco agrees to one-year deal with Browns - NBC Sports**  
Flacco returns to Cleveland after playing for the Colts in 2024.

April 11, 2025    [Read More →](#)



ABC News  
**Menendez brothers' aunt breaks her silence in exclusive interview as case goes to court - ABC News**



CNN  
**Another inflation report underlines the strength of the US economy before Trump's tariff chaos - CNN**



Politico  
**Senate confirms 'Razin' Caine to lead Joint Chiefs - Politico**  
Senators approved the unconventional pick

## 4.8 Posts

**SG Health Portal** Home Healthcare Facilities Health Blog Posts Alerts Sign In

**Health Community Posts** Create Post Reset Demo Posts

Search posts... Click to go back, hold to see history All Categories All Tags

Showing 3 posts

Wellness ♡ 15

### Tips for Staying Healthy During Flu Season

As flu season approaches, it's important to take precautions to protect yourself and your loved ones. Here are some tips: 1. Get your...

#flu #prevention #health

health\_expert Sep 15, 2023

1 comments

Environment ♡ 27

### Air Quality Chart

Understanding Air Quality Indexes

With increasing concerns about air pollution, it's essential to understand what air quality indexes mean for your health. Here's a simpl...

#air quality #health #pollution

doctor\_jen Sep 16, 2023

2 comments

Fitness ♡ 42

### Benefits of Regular Exercise

Exercise isn't just about weight management - it offers countless health benefits! Regular physical activity can: - Improve cardiovascula...

#exercise #health #fitness

wellness\_guru Sep 17, 2023

0 comments

## 4.9 Create A Post

**SG Health Portal** Home Healthcare Facilities Health Blog Posts Alerts Create Post Profile TestAccount Sign Out

**Create New Post**

Content \*  
Enter post content...

Category \*  

SC2006  
Created at 12/04/2025

Test Category  
Created at 30/03/2025

Test Category  
Created at 10/04/2025

Test Category  
Created at 10/04/2025

Test Category  
Created at 11/04/2025

Test Category  
Created at 11/04/2025

+ Create New Category

Location \*  


## 4.10 Alerts

**Singapore Health Portal**

Your comprehensive resource for healthcare information.

**Ang Mo Kio** ▼  
Sunday, 13 April 2025  
12:46 am  
Data from data.gov.sg

Humidity: 91%

**Data Layers**

- Dengue Hotspots (selected)
- Traffic Conditions
- Air Quality
- Active Alerts

Active Layer: Dengue

**Recent Alerts**

Date	Description
12 Apr 2025, 18:04	Test alert
12 Apr 2025, 18:04	Test alert
12 Apr 2025, 18:04	Test alert
12 Apr 2025, 18:04	Test alert
12 Apr 2025, 18:04	Test alert

**Weather Forecast**

Cloudy 26°C  
Rainfall: 0.1 mm  
Updated: 13/04/2025, 12:30:00 am

**Map**

**Singapore Health Portal**

Your comprehensive resource for healthcare information.

**Ang Mo Kio** ▼  
Monday, 14 April 2025  
12:02 am  
Data from data.gov.sg

Humidity: 89%

**Data Layers**

- Dengue Hotspots (selected)
- Traffic Conditions
- Air Quality
- Active Alerts

Active Layer: Dengue

**Recent Alerts**

Date	Description
13 Apr 2025, 21:58	test
13 Apr 2025, 16:59	Deadline!!!! SC2006 DEADLINE APPROACHING!!! PREPARE FOR THE DEMO!!! XD
13 Apr 2025, 15:25	Test alert

**Weather Forecast**

Cloudy 27°C  
Rainfall: 0 mm  
Updated: 13/04/2025, 11:30:00 pm

**Map**

### 3.1.3 Admin Interface

#### 4.11 Admin Profile

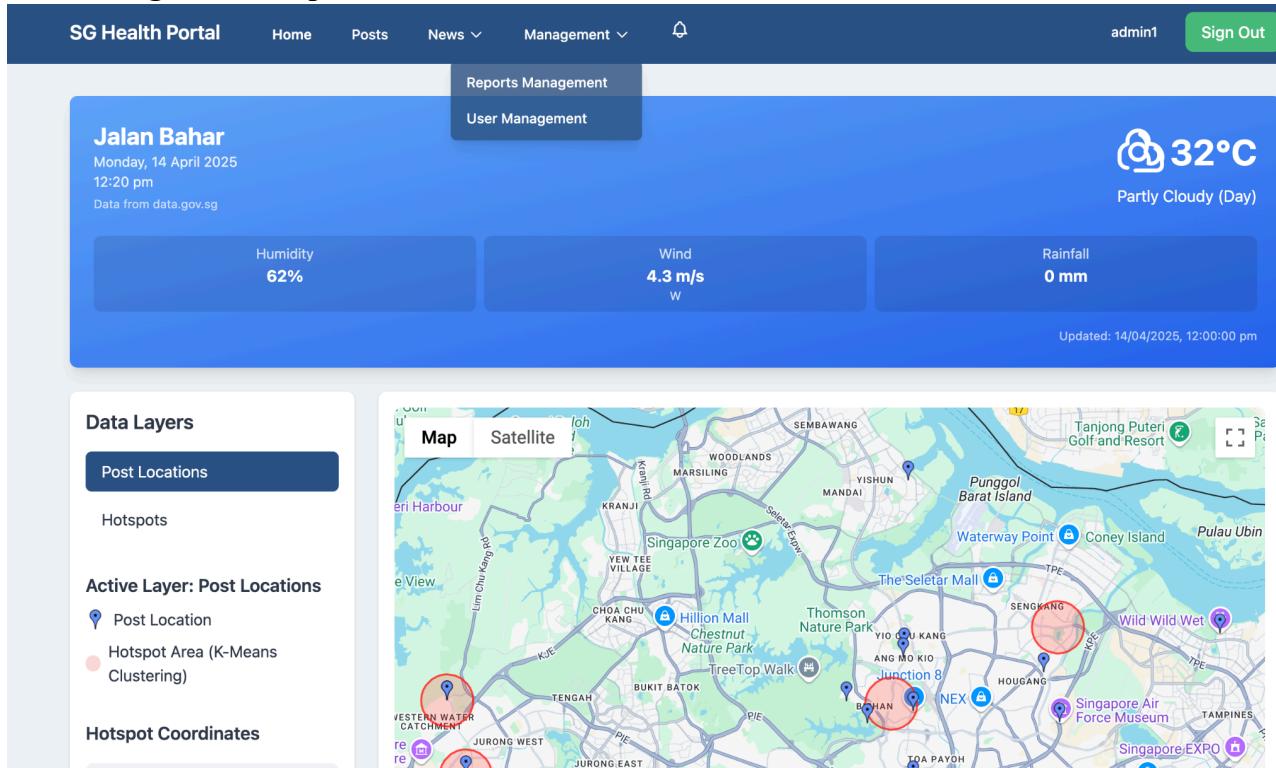
The screenshot shows the SG Health Portal Admin Profile interface. At the top, there is a dark blue header bar with the text "SG Health Portal" on the left, and "admin1" and "Sign Out" on the right. Below the header, the main content area has a light gray background. It features two main sections: "User Profile" and "Submit Feedback".

**User Profile:** This section contains "Basic Information" with fields for "Full Name" (admin1) and "User Type" (ADMIN). There is also an "Edit Profile" button.

**Submit Feedback:** This section includes a dropdown menu for "Feedback Type" set to "General Feedback", a text area for "Your Message" with placeholder text "Please share your thoughts...", and a "Submit Feedback" button.

At the bottom left of the main content area, there is a green button labeled "Change Password".

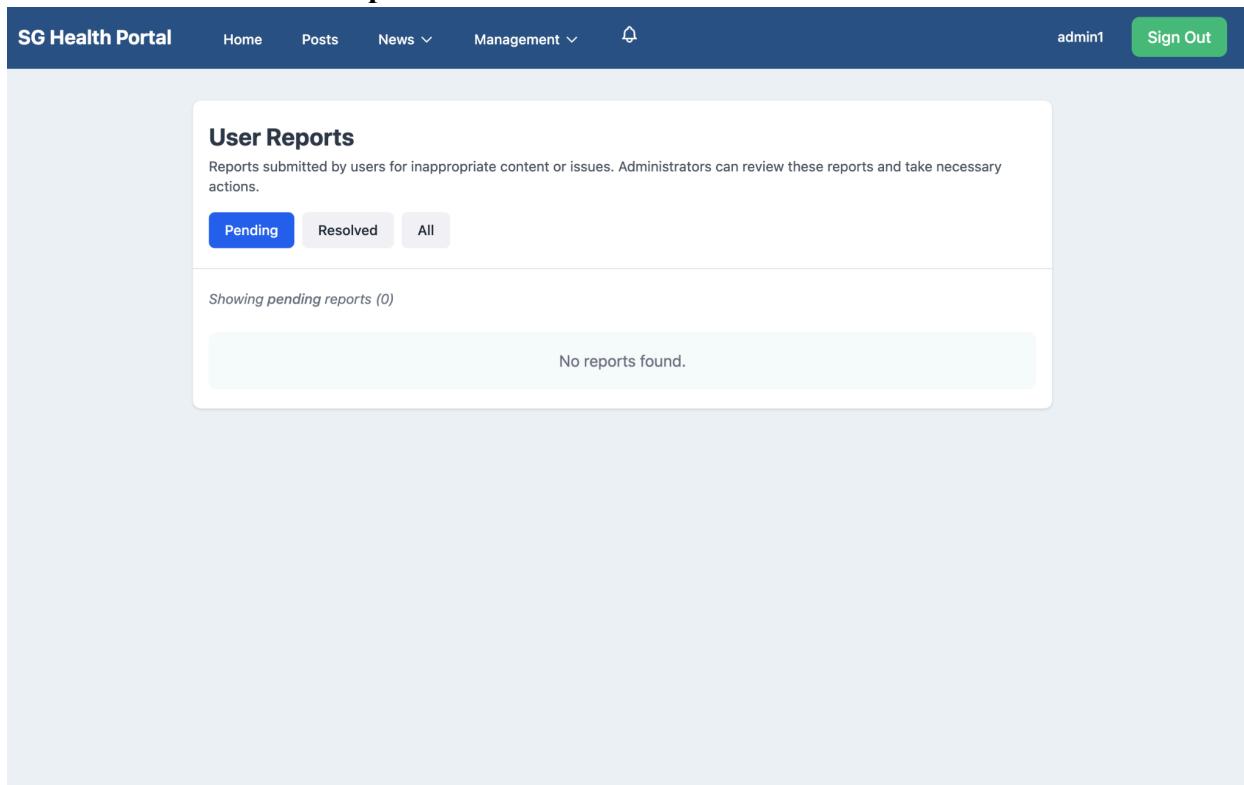
## 4.12 Management Dropdown



The screenshot shows the SG Health Portal interface. At the top, there is a navigation bar with links for Home, Posts, News, Management, and a sign-out button for 'admin1'. A dropdown menu labeled 'Management' is open, showing 'Reports Management' and 'User Management' as options. Below the navigation bar, there is a weather and environment summary for 'Jalan Bahar' on Monday, 14 April 2025, at 12:20 pm. It includes data from data.gov.sg, showing Humidity (62%), Wind (4.3 m/s W), and Rainfall (0 mm). The temperature is listed as 32°C with a partly cloudy (Day) icon. A map of Singapore is displayed, with several locations highlighted by red circles, corresponding to the 'Post Locations' layer selected in the sidebar. The sidebar also lists 'Hotspots' and 'Hotspot Coordinates'.

### 4.12.1 Reports Management

#### 4.12.1.1 No User Reports



The screenshot shows the 'User Reports' section of the SG Health Portal. The title 'User Reports' is at the top, followed by a sub-instruction: 'Reports submitted by users for inappropriate content or issues. Administrators can review these reports and take necessary actions.' Below this, there are three filter buttons: 'Pending' (highlighted in blue), 'Resolved', and 'All'. A message below the filters says 'Showing pending reports (0)'. A large text box below states 'No reports found.'

### 4.12.1.2 Pending Reports

### 4.12.1.3 Resolved Reports

### 4.12.1.4 All Reports

## 4.12.2 User Management

User	User Type	Change Type
mimic-yj	MODERATOR	MODERATOR ▾ Submit
moderate1	MODERATOR	MODERATOR ▾ Submit
moderate2	MODERATOR	MODERATOR ▾ Submit
MODERATOR Hu Han	MODERATOR	MODERATOR ▾ Submit
moderator-yj	MODERATOR	MODERATOR ▾ Submit
normal1	NORMAL	NORMAL ▾ Submit
11111111	NORMAL	NORMAL ▾ Submit
Frankhuex	NORMAL	NORMAL ▾ Submit
Karthik	NORMAL	NORMAL ▾ Submit
normal2	NORMAL	NORMAL ▾ Submit
normal4	NORMAL	NORMAL ▾ Submit
normal5	NORMAL	NORMAL ▾ Submit
NORMAL Hu Han	NORMAL	NORMAL ▾ Submit
normal-yj	NORMAL	NORMAL ▾ Submit
TestAccount	NORMAL	NORMAL ▾ Submit

## 4.13 Alerts Tab

**Jalan Bahar**  
Monday, 14 April 2025  
12:25 pm  
Data from data.gov.sg

Humidity: 61%

**32°C**  
Partly Cloudy (Day)

Rainfall: 0 mm

Updated: 14/04/2025, 12:00:00 pm

**Recent Alerts**

**Alert History** Refresh

Current Alerts: 13 | Past Alerts: 229

health Official

**Critical Health Alert**  
Water Contamination Warning: Residents near Riverbank Ave should not drink tap water until...  
14 Apr 2025, 11:55 am - Riverbank Avenue

weather Official

**Weather Warning**  
Flash Flood Risk: Due to heavy rain, avoid...  
Orchard Road until 6 PM

[View All Alerts](#)

[Send Alerts](#)

**Data Layers**

**Post Locations**

**Hotspots**

**Active Layer: Post Locations**

- 📍 Post Location
- Hotspot Area (K-Means Clustering)

**Hotspot Coordinates**

- 1.37762, 103.95435 +
- 1.34873, 103.84064 +

## 4.14 Send Alerts

SG Health Portal    Home    Posts    News    Management    admin1    Sign Out

### Create New Alert

Content \*

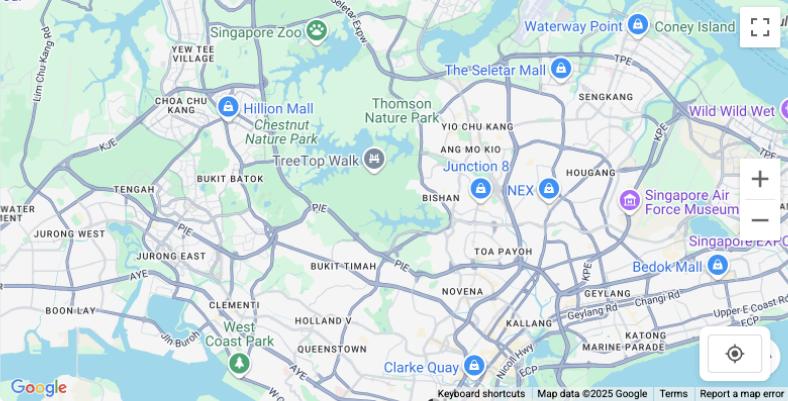
Enter post content...

Category \*

SC2006 Created at 12/04/2025	SC2006 2 Created at 14/04/2025
SC2006 good Created at 14/04/2025	test Created at 13/04/2025
Test Category Created at 30/03/2025	Test Category Created at 10/04/2025
Test Category Created at 11/04/2025	Test Category Created at 11/04/2025
Test Category Created at 10/04/2025	

+ Create New Category

Location \*



Keyboard shortcuts | Map data ©2025 Google | Terms | Report a map error

Cancel    Send Alert

### 3.1.4 Moderator Interfaces

#### 4.15 Moderator Profile

User Profile

Basic Information

Full Name: normal1

User Type: MODERATOR

Submit Feedback

Feedback Type: General Feedback

Your Message: Please share your thoughts...

Change Password

#### 4.16 User Management (Moderator)

User	User Type	Change Type
11111111	NORMAL	NORMAL ~ Submit
Frankhuex	NORMAL	NORMAL ~ Submit
Karthik	NORMAL	NORMAL ~ Submit
normal2	NORMAL	NORMAL ~ Submit
normal4	NORMAL	NORMAL ~ Submit
normal5	NORMAL	NORMAL ~ Submit
NORMAL Hu Han	NORMAL	NORMAL ~ Submit
normal-yj	NORMAL	NORMAL ~ Submit
TestAccount	NORMAL	NORMAL ~ Submit

User Management

Reports Management

User Management (Moderator)

## 3.2 Hardware Interfaces

Our platform is a web-based application and does not require any direct integration with physical hardware components. However, it is designed to run on common hardware through which users can interact with the platform.

### 3.2.1 Device Types

The application is designed to operate smoothly on various devices, including desktop computers, laptops, tablets and mobile phones, ensuring a responsive experience across different platform types.

### 3.2.2 Data and Control Interactions

Users directly interact with the platform through standard input/output hardware such as keyboards, touchscreens, mice, and device location sensors. Network interface components like Wi-Fi or mobile data are used for internet connectivity.

### 3.2.3 Communication Protocols

All client to backend communications are handled over secure HTTPS connections. The software interacts with the APIs over various web ports, ensuring encrypted and secure data transfer between components.

## 3.3 Software Interfaces

### 3.3.1 Operating Systems

The platform is capable of operating on any modern web browser including Windows, macOS, Linux, iOS, Android)

### 3.3.2 Web Browsers

The platform is capable of operating across various modern web browsers including, Chrome, Firefox, Safari, and Edge.

### 3.3.3 APIs Used

1. **Google maps API:** This API is used for location-based services such as displaying nearby hospitals/clinics, route estimations, and reverse geocoding user locations.
2. **NewsAPI:** This API is used to retrieve relevant articles and displayed in the Health Blog section of the application
3. **Data.gov.sg Environmental APIs:** A set of public APIs used to display local environmental updates that could affect public health:
  - a. Weather Forecast
  - b. Air Temperature

- c. Rainfall
- d. Relative Humidity
- e. Wind Speed and Direction

### 3.3.2 Database Used

**Safe, Smart, Strong** uses Supabase as a BaaS platform built on PostgreSQL which handles:

- 1. User authentication and session management
- 2. Storage of posts, reports, alerts, feedback, categories, and profile updates
- 3. Real-time updates for alerts systems and moderation actions

### 3.3.4 Frameworks Used

The platform is built using the tech stack:

- 1. **Vue.js**: Frontend framework used for building reactive and flexible interfaces
- 2. **TypeScript**: Provides type safety and better tooling for scalable development
- 3. **Tailwind CSS**: CSS tool used for maintaining clean, and responsive styling
- 4. **Vue Router**: Handles client-side routing between pages
- 5. **Pinia**: Used for application wide state management
- 6. **Axios**: Used to make HTTP requests for external APIs and Supabase endpoints

### 3.3.5 RESTful APIs

The application uses RESTful API interfaces for communication between frontend and backend. All API calls return and accept data formatted in JSON.

### 3.3.6 Data Sharing

All frontend and backend communication uses JSON formatting. Data persistence is handled by Supabase's real-time PostgreSQL backend.

## 3.4 Communications Interfaces

All communication between the user's browser and the backend services is conducted over HTTPS, ensuring secure, encrypted data transfer. The application uses RESTful API calls via HTTP/HTTPS for:

- 1. Retrieving news articles from NewsAPI
- 2. Retrieving environmental data from data.gov.sg APIs
- 3. Location-based services from Google Maps APIs
- 4. Interacting with Supabase backend for user authentication, database operations and storage

Both requests and responses are formatted in JSON, allowing seamless data transfer and integration with Javascript-based frontend frameworks such as Vue.js.

Supabase uses JWT (JSON Web Tokens) to handle authentications, and all authenticated API requests are secured through token-based headers.

Users access the platform through standard web browsers, which handle client-side rendering, cookie management, and session management. This arrangement ensures a responsive and consistent user experience across both desktop and mobile devices.

## 4. System Features

Please refer to Use Case Diagram and Descriptions, and Sequence Diagrams for the full, details use case diagram, use case descriptions and sequence diagrams for each use case.

### 4.1 User Authentication

#### 4.1.1 Description and Priority

Users can log in or sign up to access the **Safe, Smart, Strong**. It includes secure user authentication, encrypted password storage, and error handling of invalid attempts.

Priority: High

#### 4.1.2 Stimulus/Response Sequences

1. User navigates to the SignUp/LogIn page
2. User enters their email and password
3. The system verifies the email and password
  - a. If valid, the user is logged in and redirected to the Home page
  - b. If invalid, the user is redirected to the Home page without being logged in
4. Includes a ‘forgot password’ and signup flow

#### 4.1.3 Functional Requirements

- REQ-1: System shall authenticate users via email/password.
- REQ-2: System shall encrypt and hash passwords before storing.
- REQ-3: System shall send a password reset email if requested.
- REQ-4: System shall validate new registration data before account creation.

## 4.2 Profile Update

### 4.2.1 Description and Priority

Allows users to update their personal information securely.

Priority: High

### 4.2.2 Stimulus/Response Sequences

1. User logs in.
2. User navigates to profile settings.
3. Users make edits.
4. System validates and saves new data.

### 4.2.3 Functional Requirements

REQ-1: Only authenticated users can update their profile.

REQ-2: System shall validate all inputs.

REQ-3: Passwords must meet strength criteria.

REQ-4: System shall confirm and display profile changes.

## 4.3 Password Recovery

### 4.3.1 Description and Priority

Allows users to reset their password using their registered email.

Priority: High

### 4.3.2 Stimulus/Response Sequences

1. User clicks "Forgot Password" option on Login page
2. The system prompts users to enter their email.
3. System sends a password reset link.
4. User follows the link and sets a new password.

### 4.3.3 Functional Requirements

REQ-1: Password reset link expires after a defined period.

REQ-2: System validates email before sending reset link.

REQ-3: New password must meet password strength and security policies.

## 4.4 Create Post

### 4.4.1 Description and Priority

Users can submit community posts related to safety or health.

Priority: High

### 4.4.2 Stimulus/Response Sequences

1. User logs into their account
2. User navigates to the “Create Post”
3. The user enters required details such as content, categories, or location.
4. The user chooses to either “Publish Post” or “Cancel” the post.
5. If published, the system validates the post content for compliance with guidelines.
6. If valid, the post is stored and displayed on the community “Posts” page.

### 4.4.3 Functional Requirements

REQ-1: Only authenticated users shall be allowed to create and publish posts.

REQ-2: The system validates content before publishing the post.

REQ-3: The published posts shall have their corresponding dates on them

## 4.5 Edit/Delete Post

### 4.5.1 Description and Priority

Enables users to edit or remove their own posts. Priority: Medium

### 4.5.2 Stimulus/Response Sequences

1. The users navigate to their profile
2. The user selects a post and chooses to Edit or Delete it.
3. For edits:
  - a. The system updates the content and publishes it
  - b. The system shall re-validate the content and updates the post
4. For deletions:
  - a. The system removes the post from public view and marks it as deleted in the backend

### 4.5.3 Functional Requirements

REQ-1: System verifies ownership before allowing the user to Edit or Delete.

REQ-2: Deleted posts are no longer visible on the community post feed.

## 4.6 View Posts

### 4.6.1 Description and Priority

Displays all community posts to the user.

Priority: High

### 4.6.2 Stimulus/Response Sequences

1. The user navigates to the “Posts” page of the application.
2. The system fetches and displays posts from the database.
3. Users may search, apply filters on categories or apply tags to refine results
4. The user can scroll through posts

### 4.6.3 Functional Requirements

REQ-1: The system shall display posts in a paginated view

REQ-2: Users shall be able to search for posts by title or keywords

REQ-3: The system shall have filtering options such as categories and tags

REQ-4: If no posts match the search or filters, the system will display a “No posts found. Try adjusting your search or category filters” message

## 4.7 Send Feedback

### 4.7.1 Description and Priority

Allows users to send feedback on posts or alerts.

Priority: Medium

### 4.7.2 Stimulus/Response Sequences

1. The User navigates to their profile and clicks "Submit Feedback".
2. The user is prompted to enter comments and select a Feedback Type
3. Upon submission, the system validates the content and stores the feedback in the database.

### 4.7.3 Functional Requirements

REQ-1: The system shall display the feedback form in the user Profile page

REQ-2: The system shall provide predefined Feedback Types to classify user responses

REQ-3: The system shall store all feedback in the database.

## 4.8 Like/Dislike Post

### 4.8.1 Description and Priority

Users can like or dislike community posts.

Priority: Medium

### 4.8.2 Stimulus/Response Sequences

1. The user navigates to community “Posts” page
2. The system displays community posts with a Like and Dislike icon
3. The user can click the thumbs up or down icon to react
4. The system checks if user has reacted to the post before
  - a. If there is no prior reaction, the system records the reaction
  - b. If the same reaction is already recorded, the reaction is removed
  - c. If the opposite reaction is made, the system updates the reaction accordingly
5. The reaction count is updated in real time and displayed on the interface

### 4.8.3 Functional Requirements

REQ-1: Only authenticated users can interact with the reaction functionality.

REQ-2: The system shall prevent multiple reactions per post from the same user.

REQ-3: The system shall reflect the total like and dislike count in real-time

REQ-4: Users shall be able to remove their like and dislike from posts by clicking the same icon again

## 4.9 Report Post

### 4.9.1 Description and Priority

Users can report inappropriate or community guidelines violating posts for moderation.

Priority: High

### 4.9.2 Stimulus/Response Sequences

1. The user navigates to the community “Posts” page
2. The User selects the post they want to report and clicks the "Report" button.
3. The user selects the reason for reporting and additional details
4. The user submits the report.
5. The system stores the report and flags the post.
6. If the number of reports crosses a certain threshold, the post is hidden from public view
7. Moderators receive alerts to review the post and take action.

### 4.9.3 Functional Requirements

REQ-1: Only logged-in users can report posts

REQ-2: The system shall store the time, selected reason, and user ID of report

REQ-3: Posts with a report count past a predefined threshold shall be hidden

REQ-4: Users shall not be able to report a post more than once

## 4.10 View Alerts

### 4.10.1 Description and Priority

Real-time health and safety alerts issued by trusted authorities, and moderators and administrators are displayed

Priority: High

### 4.10.2 Stimulus/Response Sequences

1. The user navigates to the “Alers” page.
2. The system fetches current alerts in real-time from the database.
3. The system displays the alerts in chronological order of time
4. The user can select an alert to view more details

### 4.10.3 Functional Requirements

REQ-1: The system shall retrieve and display alerts as a pop-up on the user’s screen

REQ-2: The system shall update alerts in real-time

REQ-3: Alerts must include the date, time, and relevant details

## 4.11 Search Hospitals

### 4.11.1 Description and Priority

Allows users to find hospitals nearby using the interactive map.

Priority: High

### 4.11.2 Stimulus/Response Sequences

1. The user navigates to the “Healthcare Facilities” page
2. The system displays a map with hospitals and polyclinics pinpointed
3. The user can:
  - a. Select a facility marker from the map to view its name, address, number, and services
  - b. Search for a facility by name or address
  - c. Filter their search to show only Hospitals or Polyclinics
4. The system dynamically updates the map and list based on the filters applied

### 4.11.3 Functional Requirements

REQ-1: The system must use Google Maps API to display the interactive map

- REQ-2: The system must allow searching for facilities by name or location
- REQ-3: Facility details (address, name, services, and number) must be displayed when user selects facility

## 4.12 Display Alerts on Map

### 4.12.1 Description and Priority

Visual display of alerts using map.

Priority: High

### 4.12.2 Stimulus/Response Sequences

1. The user navigates to the “Home” page
2. The system displays an interactive map
3. The user can filter search by selecting a data layer (Dengue Hotspots, Traffic Conditions, Air Quality, and Active Alerts)
4. The system overlays the map with alert markers depending on level of risk
5. The user may select a marker to view details

### 4.12.3 Functional Requirements

REQ-1: The system must plot the alerts in real-time

REQ-2: The system requires Google Maps API for rendering map and placing markers

REQ-3: The system must use Singapore Government Environmental API to display alerts

## 4.13 Send Alerts

### 4.13.1 Description and Priority

Moderators can create and send verified alerts to users.

Priority: High

### 4.13.2 Stimulus/Response Sequences

1. The moderator logs in and navigates to “Send Alerts”
2. The moderator enters alert details such as content, category, and location
3. The system validates the alert and verifies moderator permissions
4. The alert is stored in the database
5. The alert is pushed as a notification to users and displayed in the “Alerts” page

### 4.13.3 Functional Requirements

REQ-1: Only authentication moderators shall be able to create and send alert

REQ-2: The system must verify the alert

REQ-2: The alerts must be logged with date, and timestamp

## 4.14 Manage Feedback

### 4.14.1 Description and Priority

Moderators and Admins can review and act on user feedback.

Priority: Medium

### 4.14.2 Stimulus/Response Sequences

1. The moderator navigates to the feedback section
2. The system displays a list of submitted feedback, including content, time, user, and type of feedback.
3. Moderator and admins review each item and can
  - a. Mark it as reviewed
  - b. Flag for followup
  - c. Delete inappropriate or spam feedback
4. The system stores the action taken by Administrator or Moderator and updates the feedback status

### 4.14.3 Functional Requirements

REQ-1: Only authenticated Administrators and Moderators may access and review user feedback

REQ-2: The system shall allow moderators to review, flag, or delete feedback

REQ-3: All moderation actions shall be logged with moderator ID, timestamp, and feedback ID.

REQ-4: Feedback moderation status shall be updated in real-time

## 4.15 Encrypt User Data

### 4.15.1 Description and Priority

Ensures encryption of passwords and sensitive data.

Priority: High

### 4.15.2 Stimulus/Response Sequences

1. User registers or updates personal information
2. The system validates the data
3. The system encrypted the information before storing it in the database
4. The information is then stored in the database

5. During login, the entered password is encrypted

#### **4.15.3 Functional Requirements**

- REQ-1: The system shall use one-way hashing for passwords.  
REQ-2: No sensitive information shall be stored in plaintext

### **4.16 Manage User Accounts**

#### **4.16.1 Description and Priority**

Admins can suspend, delete, or restore accounts.  
Priority: High

#### **4.16.2 Stimulus/Response Sequences**

1. The admin logs in and navigates to the “User Management” page
2. The admin searches for the user and selects the user account
3. The admin chooses action to perform on the user account.
4. The system verifies admin permissions and updates status.

#### **4.16.3 Functional Requirements**

- REQ-1: Actions logged with admin ID.  
REQ-2: User is notified of changes.

### **4.17 Manage Moderators**

#### **4.17.1 Description and Priority**

Admins can update moderator roles  
Priority: Medium

#### **4.17.2 Stimulus/Response Sequences**

1. Admin views list.
2. Edits permissions.
3. Save changes.

#### **4.17.3 Functional Requirements**

- REQ-1: Requires admin authentication.  
REQ-2: All role changes logged.

## **4.18 Manage Reports and Post Removals**

### **4.18.1 Description and Priority**

Admins oversee moderation decisions.

Priority: High

### **4.18.2 Stimulus/Response Sequences**

1. Admin reviews reports.
2. Confirms or overrides moderation.

### **4.18.3 Functional Requirements**

REQ-1: Appeals reviewed with evidence.

REQ-2: System updates post status accordingly.

## **4.19 Manage Maintenance**

### **4.19.1 Description and Priority**

Admins configure system settings and maintenance.

Priority: High

### **4.19.2 Stimulus/Response Sequences**

1. Admin opens settings.
2. Edits and saves.
3. Users are notified if system downtime is scheduled.

### **4.19.3 Functional Requirements**

REQ-1: Validate inputs before applying.

REQ-2: Users notified of maintenance.

## **4.20 Sending Notifications and Alerts**

### **4.20.1 Description and Priority**

Admins can send system-wide messages.

Priority: High

### **4.20.2 Stimulus/Response Sequences**

1. Admin submits notification.
2. System validates and sends.

### **4.20.3 Functional Requirements**

REQ-1: All notifications must be logged.

REQ-2: Secure sending to authenticated users.

## **4.21 Review and Manage API Integrations**

### **4.21.1 Description and Priority**

Admins configure and monitor external APIs.

Priority: High

### **4.21.2 Stimulus/Response Sequences**

1. Admin selects API.
2. Updates keys/settings.
3. System saves and verifies.

### **4.21.3 Functional Requirements**

REQ-1: Validate API key format.

REQ-2: Store changes with admin log entry.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

1. The platform should load all primary interface screens within 10 seconds
2. The system must be able to handle at least 500 simultaneous users without notable performance degradation
3. The application should retrieve and process the user's location within 3 seconds of permission granted

### 5.2 Safety Requirements

1. The system must not disclose sensitive user information (e.g. location or identity) without permission
2. Alerts must be verified prior to being sent out to avoid spread of misinformation

### 5.3 Security Requirements

1. All passwords must be hashed before being stored in the database
2. All communication must be conducted over HTTPS
3. Use JWT for all authenticated requests

### 5.4 Software Quality Attributes

1. **Commented Code:** All frontend and backend source codes will be accompanied by concise comments, ensuring interpretability, supports future maintenance, and allows code reusability
2. **API Documentation:** Backend API endpoint implementation is accompanied with documentation to indicate endpoint URLs, required request body, and expected response body
3. **User Feedback and Testability:** The system shall display appropriate messages when actions are completed by users, to assist in debugging and improve testability.
4. **Reliability:** The application should fail to load or crash at most 1% of the time
5. **Maintainability:** The codebase must follow modular and reusable patterns to support ease of maintenance
6. **Scalability:** The system architecture shall support increased user traffic and growth without a notable performance degradation

## 5.5 Business Rules

The Business Rules provide a framework that governs how the platform operates, user roles, and system behaviours. This will ensure that **Safe, Smart, Strong** runs securely, and efficiently.

### 5.5.1 User Access and Permissions

1. Only authenticated users can create, edit, delete, or report posts
2. Restricted users can only view publicly available content, but are unable to interact with them
3. Passwords must meet minimum strength criteria
4. After 5 failed login attempts, the user account will be temporarily locked

### 5.5.2 Moderation and Reporting

1. Posts that receive 3 or more reports are hidden and awaiting moderator review
2. Moderators are responsible of reviewing feedback and reported content in alignment with platform guidelines
3. All moderator actions must be logged with timestamps for traceability and audit purposes

### 5.5.3 Administrator Authority

1. Admins are able to suspend, restore, or permanently delete user accounts based on violations or appeals
2. Admins have full access to override moderator decisions and manage platform configurations
3. Only admin is able to manage user roles and assign/remove moderator privileges

### 5.5.4 Data Integrity and Communication

1. All feedback, alerts, and posts are time-stamped and stored securely in the database
2. All API communication must be over HTTPS with proper authentication tokens
3. Alert messages sent out must be verified by moderators or admins before dispatch

### 5.5.5 Operational Standards

1. The platform will operate at all times, with moderators/admins responsible for managing real-time alerts and user generated content
2. In cases of emergency system maintenance, notifications must be sent out to users 24 hours in advance
3. All location-based services must request user permissions and comply with privacy and data protection standards, ensuring no location data is accessed or stored without permission

## 6. Other Requirements

### 6.1 Database Requirements

- The application uses Supabase for the backend-as-a-service platform, which incorporates PostgreSQL as a database
- All user information, posts, alerts, and feedback is stored in structures tables
- Password must be hashed, and therefore stored in an encrypted format
- The database must support real-time updates for alerts and blog posts
- All database operations must be logged for audit purposes

### 6.2 Internationalisation

- The system, initially, only supports English as the primary language. In the future, releases will support multiple languages.
- All text content must be stored in a way that maintains ease in translation in the future

### 6.3 Legal Requirement

- The platform must comply with PDPA (Personal Data Protection Act) Singapore, ensuring that all user information is protected, not shared, and can be deleted upon request
- Use of third-party APIs must comply with their respective terms of service and usage limits

### 6.4 Reusability

- All components must be designed using reusable Vue components to improve maintainability
- Functions like API calls should be contained within service or utility modules to allow reusability across the app

### 6.5 Accessibility Requirements

- The platform should follow the WCAG 2.1 standards for web accessibility

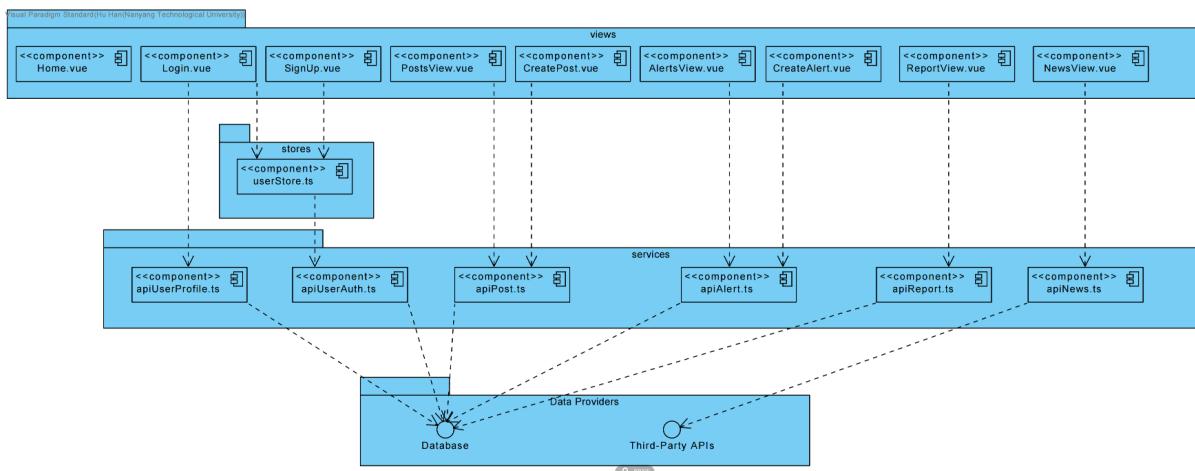
## Appendix A: Glossary

Term	Definition
SRS	Software Requirements Specification - A document that details the functionality and expected behaviour of the software
Safe, Smart, Strong	The name of the web application built by Team 1, which allows users to access health and safety resources and community posts in Singapore
Supabase	An open source backend-as-a-service (BaaS) platform used for database management, authentication, and API hosting
Vue.js	A progressive javascript framework for building the frontend user interface of a software application
TypeScript	An open source high-level programming language, superset of javascript that adds static type checking to improve code quality and maintainability
Tailwind CSS	A utility-first CSS framework which is used to build custom designs on frontend components
API	Application Programming Interface - A set of rules and protocols that allow different software applications to communicate and interact with each other
JSON	Javascript Object Notation - A data format used for storage and exchanging data between client and server
JWT	JSON Web Token - A stateless method of securely transmitting data between parties as a JSON object, used for authentication
PDPA	Personal Data Protection Act - Singapore's data protection law that governs the collection, use, and disclosure of personal data
WCAG 2.1	Web Content Accessibility Guidelines - A set of internationally recognised guidelines designed to make web content accessible for people with disabilities
Frontend	The part of the application that the user interacts with directly via the browser

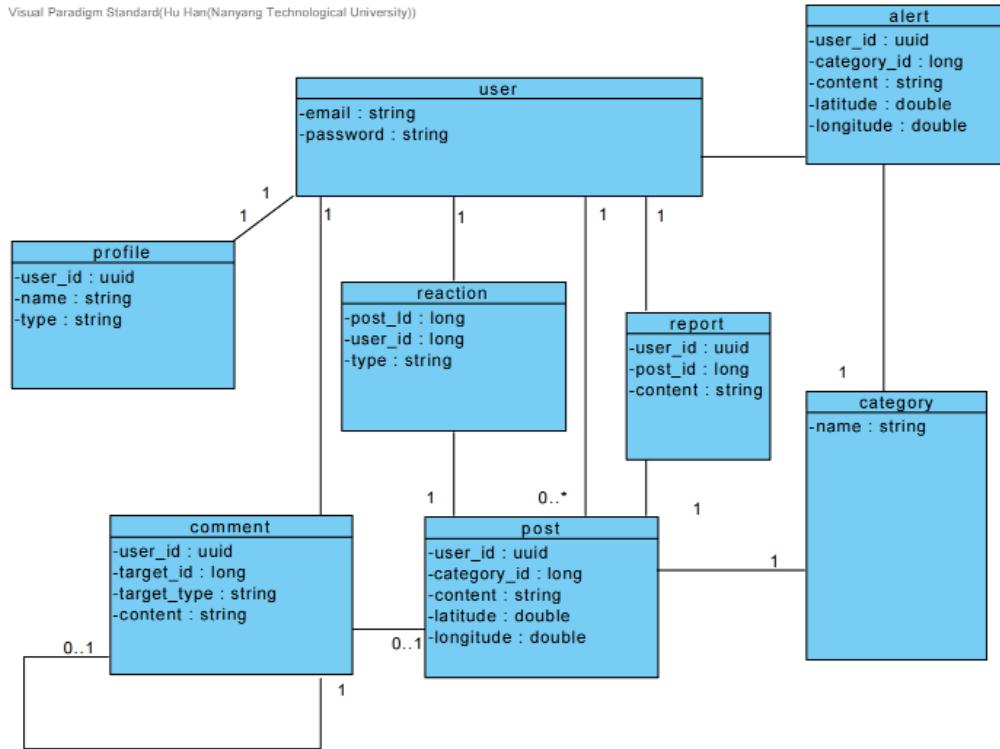
Backend	The server side logic, databases, and architecture that powers an application
---------	---

## Appendix B: Analysis Models

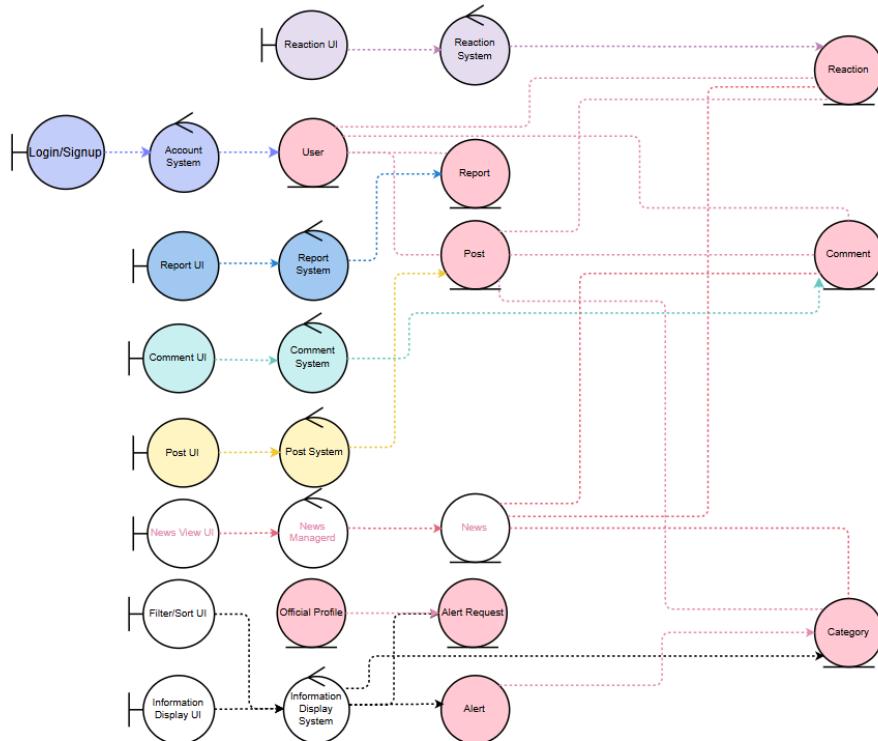
### System Architecture Diagram



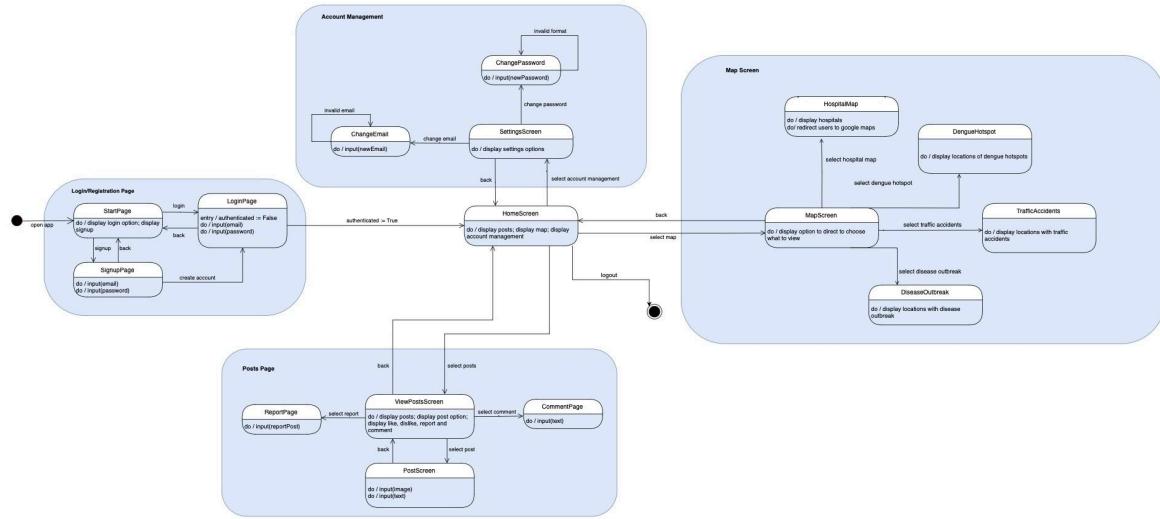
### Class Diagram



## Boundary and Control Classes



## Dialog Map



## Appendix C: To Be Determined List

Source: [http://www.frontiernet.net/~kwiegers/process\\_assets/srs\\_template.doc](http://www.frontiernet.net/~kwiegers/process_assets/srs_template.doc)