

**COLLEGE OF ENGINEERING**

**SCHOOL OF COMPUTER SCIENCE & ENGINEERING**

**CZ3003: Software System Analysis & Design   
Academic Year 2016/17**

**Project Documentation**

Revision Date: 24/10/2016

Version: 1.3

Students’ Name: Thomas Lim Jun Wei (U1521407L)

Lim Zi Yang (U1522218E)

Huang Jian Wei (U1521567A)

Yong Guo Jun (U1440217C)

Lim Hao Zhe (U1521589B)

Sim Long Siang (U1522053H)

Genevieve Lam (U1521863H)

Goh Ka Hian (U1520081B)

Lab Group: SSP3

**Table of Contents**

[1. Introduction](#_gjdgxs)

[1.1. Product Description](#_30j0zll)

[1.1.1. Purpose](#_1fob9te)

[1.1.2. Scope of the System](#_3znysh7)

[1.1.3. Users and Stakeholders](#_2et92p0)

[1.1.4. Assumptions](#_tyjcwt)

[1.2. Product Features](#_1t3h5sf)

[1.2.1. Incident Management](#_4d34og8)

[1.2.2. Display Incident Status](#_2s8eyo1)

[1.2.3. Communications](#_17dp8vu)

[1.2.4. Assets Tracking](#_3rdcrjn)

[1.2.5. Update Live Information](#_26in1rg)

[1.3. User Classes and Characteristics](#_lnxbz9)

[1.3.1. Staffs](#_35nkun2)

[1.3.2. Appointment holder of Prime Minister Office (PMO)](#_1ksv4uv)

[1.3.3. Call Centre Operators](#_44sinio)

1.3.4. Government Agencies

[1.4. Operating Environment](#_2jxsxqh)

[1.5. Design and Implementation Constraints](#_z337ya)

[2. Use Case Diagram](#_3j2qqm3)

[3. Use Case Description](#_1y810tw)

[3.1. Monitor Incident](#_4i7ojhp)

[3.2. Command Asset](#_2xcytpi)

[3.3. PMO Status](#_1ci93xb)

[3.4. Manage Crisis](#_3whwml4)

[3.5. Notify Public](#_2bn6wsx)

[4. User Documentation](#_qsh70q)

[4.1. Assumptions and Dependencies](#_3as4poj)

[4.2. External Interface Requirements](#_1pxezwc)

[4.2.1. User Interfaces](#_49x2ik5)

4.2.1.1 Login Page

4.2.1.2 Home Page

4.2.1.3 Log Incident

4.2.1.4 Edit Incident

4.2.1.5 View Incident

4.2.1.6 View Crisis

4.2.1.7 View Assets

4.2.1.8 Add Assets (SCDF/NEA/SPF/SAF)

4.2.1.9 Delete Users

4.2.1.10 Display Report

4.2.1.11 Notify Public

4.2.1.12 Email PMO

4.2.1.13 View Email

4.2.1.14 Users Scope

[4.2.2. Hardware Interfaces](#_2p2csry)

[4.2.3. Software Interfaces](#_32hioqz)

[4.2.4. Communication Interfaces](#_1hmsyys)

[5. System Features](#_19c6y18)

[5.1. Functional Requirements](#_3tbugp1)

[5.2. Software Requirements](#_1302m92)

[5.3. User Interface Requirements](#_3mzq4wv)

[6. Other Non-Functional Requirements](#_2250f4o)

[6.1. Performance Requirements](#_haapch)

[6.2. Security Requirements](#_319y80a)

[6.3. Extensibility Requirements](#_1gf8i83)

[6.4. Software Quality Attributes](#_40ew0vw)

7.     Appendix A - Glossary

8.     Appendix B - Analysis Models

[8.1. Class Diagram](#_upglbi)

[8.2. Context Diagram](#_3ep43zb)

[8.3. Data Flow Diagram](#_1tuee74)

[8.4. Dialog Map](#_2szc72q)

[8.5.](#_184mhaj) [Architecture Diagra](#_184mhaj)m

8.6. Entity-Relational (ER) Diagram

9. Appendix C – Policy Definition

10.  Appendix D – Meeting Minutes

# Introduction

## Product Description

### Purpose

The purpose of this document is to describe the requirements specification for C.R.U.X, the Crisis Management System (CMS), a system where it allows collaboration between government agencies and notifying the public in time of emergency.

### Scope of the System

C.R.U.X serves as a platform to allow collaboration between government agencies in times of occurrence of crises (Fire/Flood). It not only allows crisis handling by providing immediate responses and assistance to resolve the crisis, it also shows the transition from an incident to a crisis when an incident had gone beyond the required safety measures (Refer to Policy Definition), and the transition when a crisis transiting to an incident whenever a crisis is no longer in the alert zone levels (Refer to Policy Definition). When an incident is being reported by a member of public, call centre operators will lodge the incident’s information into C.R.U.X and monitor if it goes beyond the safety alert level. Given the case when it had surpassed the alert level, assets will be dispatched from relevant agencies to deal with the crisis. In the midst of the resolving the crisis, key decision makers will be able to observe the status changes, and notify the PMO via email as well as the public via social media channels (Twitter, Facebook, or SMS).

### Users and Stakeholders

Users of C.R.U.X includes the government agencies, call centre operators, key decision makers and ministers. Government agencies include the Singapore Police Force (SPF) and Singapore Civil Defence Force (SCDF). Key decision makers include Prime Minister’s Office. As for the members of public, they will be indirectly using the system as they will receive notifications as a form of safety awareness whenever there is occurrence of crises, to aid them in making better decision with the situations they are facing.

### Assumptions

* Assumptions are made such that relevant assets must be dispatched whenever incidents had surpassed the safety alert levels, becoming a crisis.
* The Prime Minister's office shall receive a status report summarizing key indicators and trends every 30 minutes.
* C.R.U.X is always available 24/7 in times of crisis, meaning to say appropriate actions will also be executed even in the non-office hours.
* Users of C.R.U.X should be well-trained in using the system.

## 1.2. Product Features

The product features describe the interactions between the users and the system that performs the necessary activities to map the user inputs to the desired outputs. It documents what the system mustbe able to perform.

### Incident Management

C.R.U.X allows call operators to log an incident when an incident happens and subsequently monitor the situation. The system will send a notification to commanders of related emergency agencies informing them of the current situation. An incident can be escalated into “Crisis” status if it hits any requirement stated in our policy definition. If the incident/crisis is determined to be over, call operators can choose to delete incident.

### Display Incident Status

Once created, users are able to view the situation of the incident. These included the number of casualties, area and infrastructure affected. It must be updated in real time. At the same time, a map will be marked with a marker showing the location of the incident.

### Communications

C.R.U.X has an in-built communication platform that comes in the form of a chat room to further facilitate different government agency in sharing their resource and plans to tackle the crisis.

### Assets Tracking

C.R.U.X will also provide the numbers and location of the manpower and personnel deployed to handle the crisis from different emergency agencies. This is shown through a different coloured marker marked on the map.

### Update Live Information

C.R.U.X allows different government bodies to share their information and then post notification to the public through social media update such as Twitter and Facebook as a single entity. Also, SMS will be sent out to the members of public. This allows information feeds to be publish frequently so that public will be better notified on the current situation crisis.

## User Classes and Characteristics

C.R.U.X is meant to offer a platform for collaboration between various government agencies. The system has to be time-critical and its user must be well equipped with knowledge and technical skills on how to handle the system. The users are divided into three categories, mainly the staffs, key appointment holders of Prime Minister Office (PMOs) and lastly the call centre operators.

### Staffs

Staffs are mainly classified into three group, normal staff, head of staff and public relations (PR) manager. All staffs are able to view and monitor incidents using the system. Only head of staff is able to delete incident. Only PR managers are able to update social media such as Facebook page. Staff are required to handle the troubleshooting and management of the system.

### Appointment holder of Prime Minister Office (PMO)

Appointment holders of PMO are basically cabinet ministers and key decision makers. These users do not have to know how the technical aspect of the system. They mainly monitor the situation of the crisis using the real-time map and also make decisions based on the current crisis level.

### Call Centre Operators

Call Centre operators are users who create incident using the system. They must be trained to use the system in an efficient manner.

**1.3.4. Government Agencies**

Government Agencies include National Environment Agencies (NEA), Singapore Police Force (SPF), Singapore Armed forces (SAF) and also Singapore Civil Defense Force (SCDF).

Each agency is required to update their assets in the event they are sent to handle an incident or a crisis.

**1.3.5. Public**

Public are considered passive users are they do not directly interact with C.R.U.X. They will just receive updates through social media platform such as Facebook, Twitter or through SMS.

## Design and Implementation Constraints

During a crisis, every second counts. C.R.U.X is a time critical system that must be available 100% of the time. It is a necessity to ensure constant reliability hence there must be not be any failure or downtime. As such we must take redundancy into account to improve reliability and quality assurance of the system. We must also take into account the interface implementation such that users are able to take the minimal steps to achieve their objectives. This in terms speed up the efficient of countermeasures to handle the crisis.

As is it hosted on a web server, security is also one our major concern. We must ensure that our website is not prone to malicious cyber-attacks.

C.R.U.X is developed on a Laravel framework, hence developers will have to be familiar with PHP and html programming language and familiar with their syntax. Furthermore, as Facebook and Twitter are external components, C.R.U.X does not have control over it.

# Use Case Diagram

# 

# Use Case Description

## 3.1 Monitor Incident

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | C.R.U.X Crisis Management System | | |
| Use Case Name: | Monitor Incident | | |
| Created by: | Guo Jun | Last Updated By: | Long Siang |
| Date Created: | 02 Sep 2016 | Date Last Updated: | 11 Nov 2016 |
| Actor: | Government Agency (GA) | | |
| Description: | The user will monitor all incidents and assets sent to affected location | | |
| Preconditions: | The user must have accessed the CMS. | | |
| Postconditions: | - | | |
| Priority: | High | | |
| Frequency of Use: | Frequent | | |
| Flow of Events: | 1. User clicks on “View Incident” on the “Main Page”. 2. System displays the list of all incidents and its respective details. 3. User clicks on “View Map” 4. System displays the map that consists of different coloured markers:   - Red = Crisis  - Yellow = Fire Incident  - Blue = Flood Incident  5. User mouse over one marker.  6. Marker will float a text box stating the description of the incident/Crisis | | |
| Alternative Flows: | AF1 - If the user received an incident to be logged  1. The user clicks on “Log Incident”  2. System displays a “Create New Incident” form  3. User enters the following information:  - Case ID  - Incident Name  - Operator Name  - Incident Location  - Incident Timing  - Incident Description  - Remarks  4. User selects an option from the following:  - Alert Level  - Category of Incident (Fire/Flood)  5. Given the longitude and latitude, the user enters the value and a marker will be placed on the map denoting the place of incident.  6. User clicks “Create”  7. System will log all the information entered  8. System will place the correct coloured marker according to the category of incident and the alert level.  9. User clicks on “Home page”  10. Return back to “Flow of Event – Step 1”  AF2 - If the user is required to edit an incident  1. User clicks on “Edit Incident”  2. System displays a list of incident available  3. User clicks on “Update” button for the incident to be edited  4 System displays the information form of the selected incident  5. User edits any textbox as required  6. User clicks on “Update”  7. System will change any information for the selected incident  8. Return back to “Flow of Event – Step 1” | | |
| Exceptions: | EX1 – If an incident has been handled and to be deleted  1. User clicks on “delete”  2. System will remove the deleted incident from the list  3. System will remove any markers relating to the deleted incident | | |
| Includes: | - | | |
| Special Requirements: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

## 3.2 Command Asset

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | C.R.U.X Crisis Management System | | |
| Use Case Name: | Command Asset | | |
| Created by: | Guo Jun | Last Updated By: | Long Siang |
| Date Created: | 02 Sep 2016 | Date Last Updated: | 11 Nov 2016 |
| Actor: | Emergency Service Agencies (SAF, SCDF, SPF, NEA) | | |
| Description: | The user will dispatch relevant asset for different crisis. | | |
| Preconditions: | The user must have accessed the CMS. | | |
| Postconditions: | The user will resolve the crisis fully or attempt to mitigate the seriousness of the crisis (transiting to incident). | | |
| Priority: | High | | |
| Frequency of Use: | Frequent | | |
| Flow of Events: | 1. User clicks on “View Asset” on the “Main Page”. 2. System will display a list of asset being dispatched. 3. User clicks on “View Map” 4. System will display a map showing Green coloured markers, denoting the current position of the asset. 5. User mouse over to one of the green coloured marker 6. Marker will float a text box stating the description of the asset | | |
| Alternative Flows: | AF1 - If the user wants to dispatch an asset  1. User clicks on “Add assets”  2. System displays a “Add Asset” form  3. User enters the following information:  - Asset ID  - Asset Name  - Asset Type  - Asset Location  - Asset Description  - Asset Status  - Asset Remarks  4. User selects an option from the following:  - Asset Department  5. Given the longitude and latitude, the user enters the value and a marker will be placed on the map denoting the current position of the asset.  6. User clicks on “Add”.  7. System will log all the information entered  8. System will place the green coloured marker on the location specified.  9. Return back to “Flow of Event – Step 1” | | |
| Exceptions: | EX1 – If situation has been handled and asset is to be deleted  1. The user clicks on “View Asset”  2. The user clicks on “delete” button of the asset to be deleted  3. The system will remove the deleted incident from the list  4. The system will remove any markers relating to the deleted incident | | |
| Includes: | - | | |
| Special Requirements: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

## 3.3 Receive Status Report

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | C.R.U.X Crisis Management System | | |
| Use Case Name: | Receive Status Report | | |
| Created by: | Guo Jun | Last Updated By: | Long Siang |
| Date Created: | 02 Sep 2016 | Date Last Updated: | 08 Nov 2016 |
| Actor: | Prime Minister’s Office (PMO) | | |
| Description: | The user will receive a status report of the crisis that is happening ongoing from the system | | |
| Preconditions: | The user must have access to his/her own email account and CMS. | | |
| Postconditions: | PMO will receive an email that consists the updates of the most updated incident/crisis | | |
| Priority: | High | | |
| Frequency of Use: | Frequent | | |
| Flow of Events: | 1. System will display a “Create Email” form on the “Home Page”. 2. User enter information for the following:   - Subject  - Operator Name  - Message   1. User clicks on “Send” 2. System will send the Email to PMO office that is initially pre-inserted into the system. | | |
| Alternative Flows: | - | | |
| Exceptions: | - | | |
| Includes: | - | | |
| Special Requirements: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

## 3.4 Manage Crisis

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | C.R.U.X Crisis Management System | | |
| Use Case Name: | Manage Crisis | | |
| Created By: | Ka Hian | Last Updated By: | Long Siang |
| Date Created: | 02 Sep 2016 | Date Last Updated: | 08 Nov 2016 |
| Actor: | Emergency Services (SAF, SCDF, SPF, NEA) | | |
| Description: | The user will monitor all crisis and determine the severity of it. | | |
| Preconditions: | The user must have accessed to the CMS. | | |
| Postconditions: | The user will be able to perform appropriate actions when an incident became a crisis.  The user will also make sure incidents that are transited from crises previously, are monitored. | | |
| Priority: | High | | |
| Frequency of Use: | Frequent | | |
| Flow of Events: | 1. User click on “View Crisis” on the “Main Page”.  2. System displays a list of Crisis with their date and summary information.  3. User clicks on “View Map”  4. System displays the map that consists of red coloured markers denoting the crisis  5. User mouse over one marker.  6. Marker will float a text box stating the description of the crisis | | |
| Alternative Flows: | AF1 – If the user wants to escalate an incident to a crisis  1. User clicks on “View Incident”.  2. System displays a list of current incidents.  3. User clicks on “escalate” button on the particular incident  4. System will remove the escalated incident from the incident list  5. System will insert the escalated incident into the crisis list  6. Return to “Flow of Events – Step 1” | | |
| Exceptions: | EX1 – If the user wants to de-escalate a crisis to an incident  1. User clicks on “View Crisis”.  2. System displays a list of current crisis.  3. User clicks on “de-escalate” button on the particular crisis  4. System will remove the escalated crisis from the crisis list  5. System will insert the escalated crisis into the incident list  EX2 – If the user wants to change the threat condition (Alert Level) of the crisis  1. User click on “View Crisis” on the “Main Page”.  2. System displays a list of Crisis with their date and summary information.  3. User choose an alert level from the drop-down box of the selected crisis  4. User clicks on “Update”.  5. System will refresh and change the alert level to the user’s selection. | | |
| Includes: | - | | |
| Special Requirements: | - | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

## 3.5 Notify Public

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | C.R.U.X Crisis Management System | | |
| Use Case Name: | Notify Public | | |
| Created By: | Ka Hian | Last Updated By: | Long Siang |
| Date Created: | 02 Sep 2016 | Date Last Updated: | 08 Nov 2016 |
| Actor: | Government Agency | | |
| Description: | Allows user to notify the public via the web application. | | |
| Preconditions: | User must have accessed to the CMS | | |
| Postconditions: | The actor made a social media post announcement to the public. | | |
| Priority: | Normal | | |
| Frequency of Use: | Frequent | | |
| Flow of Events: | 1. User clicks on “Notify Public” on the “Home Page” 2. System displays 3 textboxes for 3 separate social media platform:   - Facebook  - Twitter  - SMS  3. For each of the social media platform, the user will type in the post he/she wants to broadcast to the public.  4. User clicks on “Publish”/” Send” button.  5. System prompts for confirmation of publishing/sending.  6. User click “OK”.  7. System will process the message typed by user and publish it to the respective social media page. (For Facebook and Twitter)  8. System will send the message typed by user to public. (SMS) | | |
| Alternative Flows: | - | | |
| Exceptions: | - | | |
| Includes: | - | | |
| Special Requirements: | S-1: For the SMS notification, only users subscribed to the system will receive the announcement | | |
| Assumptions: | - | | |
| Notes and Issues: | - | | |

# User Documentation

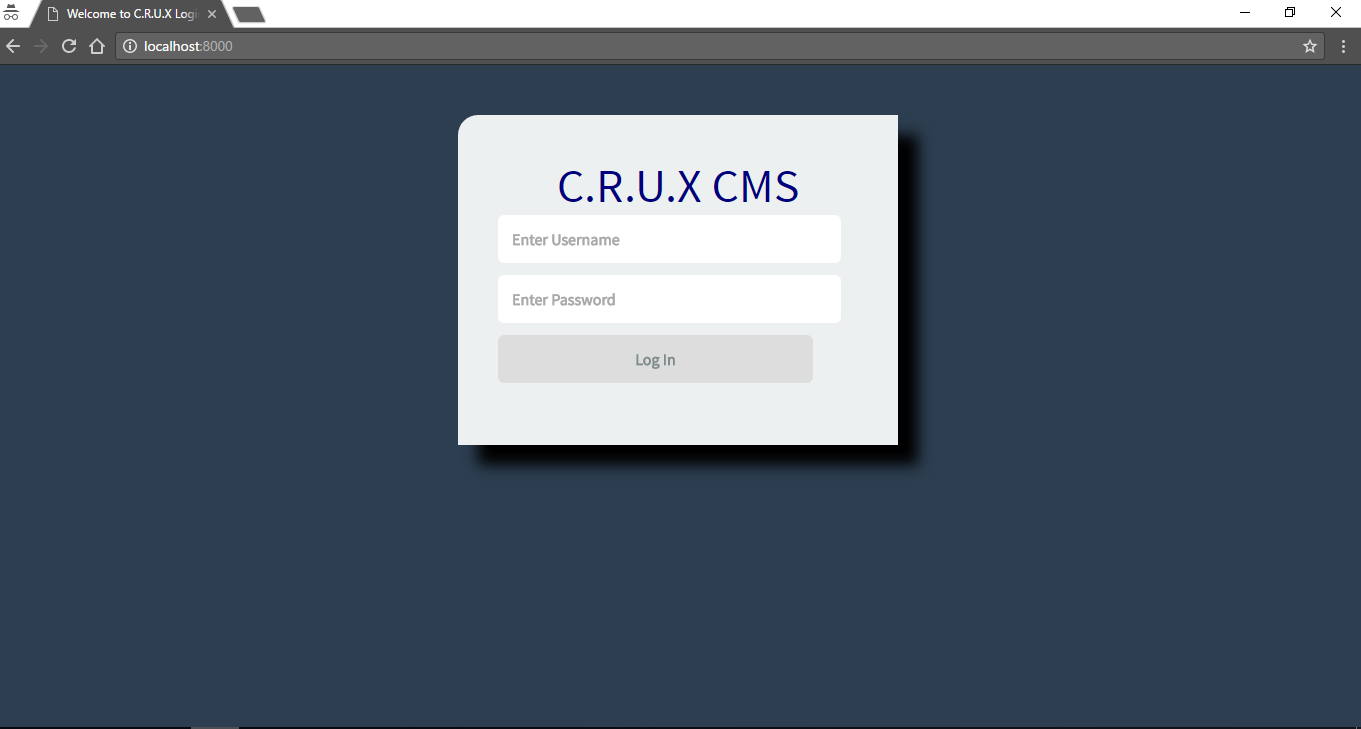
## 4.1 Assumptions and Dependencies

* Assumptions are made such that relevant assets must be dispatched whenever incidents had surpassed the safety alert levels, becoming a crisis.
* The Prime Minister's office shall receive a status report summarizing key indicators and trends every 30 minutes.
* C.R.U.X is always available 24/7 in times of crisis, meaning to say appropriate actions will also be executed even in the non-office hours.
* Users of C.R.U.X should be well-trained in using the system.
* Twitter and Facebook API must be online to post incident and crisis information.
* SMS API gateway must be available to send to the members of public.
* Crisis may or may not be caused by an incident. Given the fact that if an incident is transiting to a crisis, it surpassed a safety alert level. Otherwise, it occurs as a crisis right from the start.

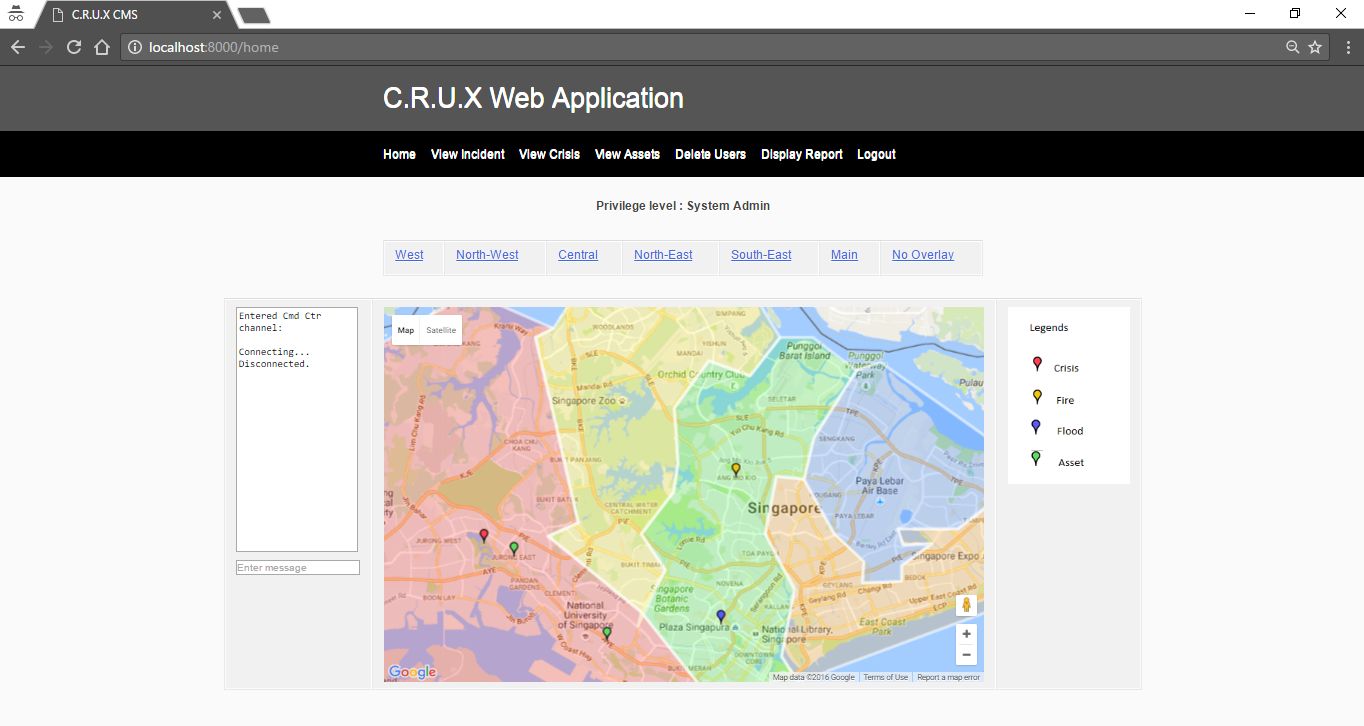
## 4.2 External Interface Requirements

### 4.2.1 User Interfaces

C.R.U.X is a web application that conform to W3C standards. As C.R.U.X is an application for private use, all users will be trained for day-to-day usage of their respective scope. C.R.U.X is also a time-critical application; hence user interface must be as concise as possible.

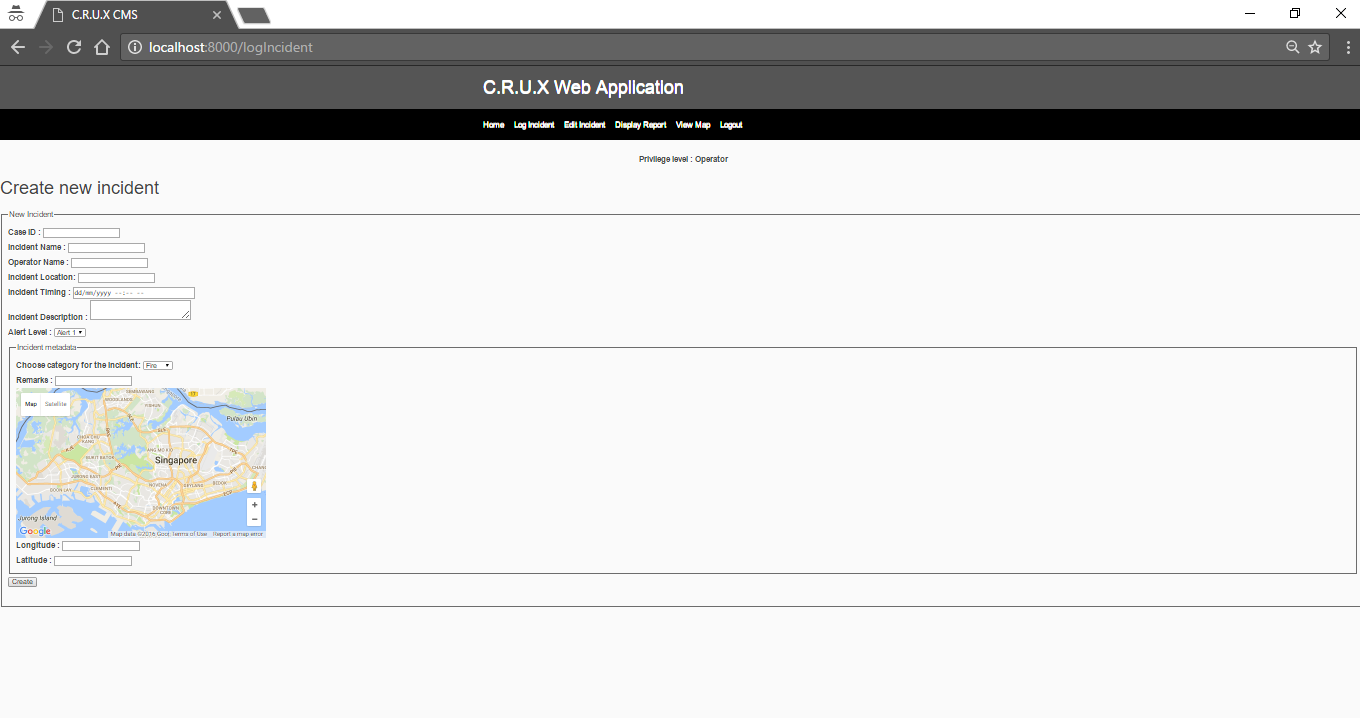
**4.2.1.1 Login Page**

**4.2.1.2 Home page (Dashboard for different users may vary)**



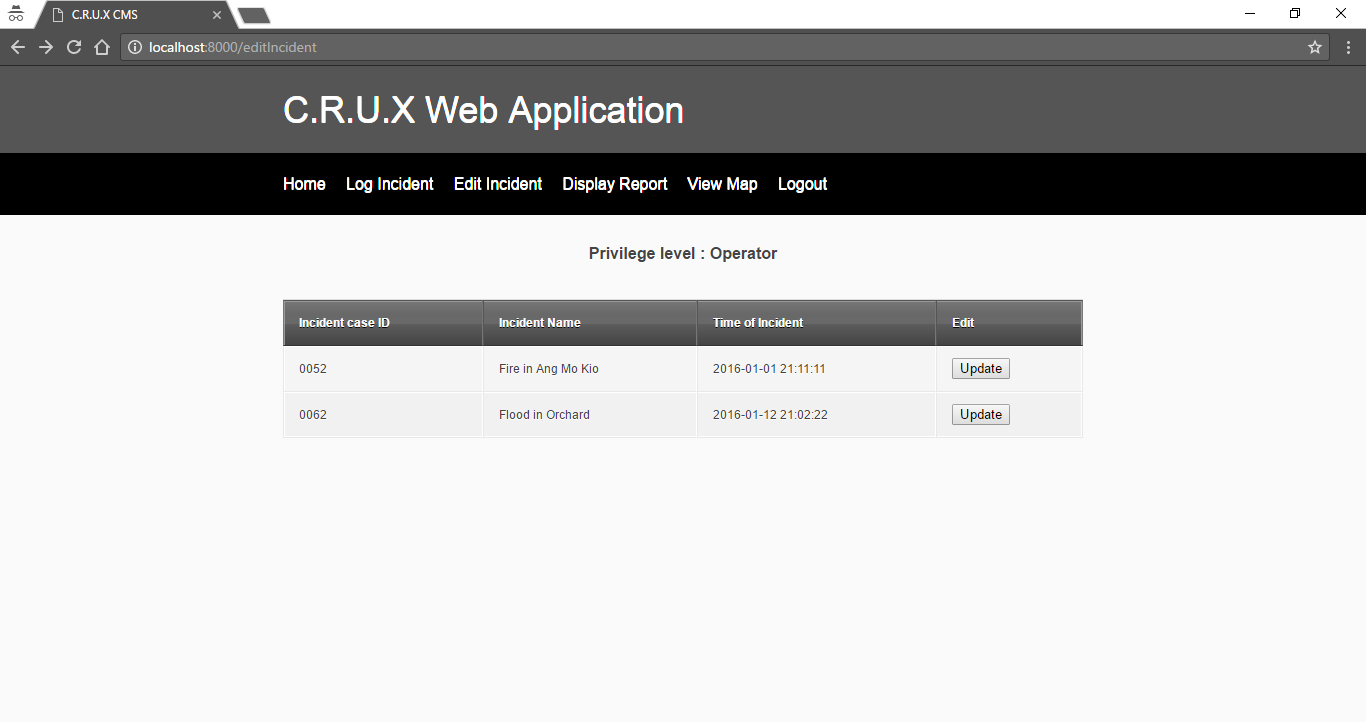
In the home page, user is able to view the Singapore map with markers that indicate incidents, crisis and assets deployed around Singapore. Commanders are also able to chat among each other using the chat provided at the left side of the page.

**4.2.1.3 Log Incident**

****

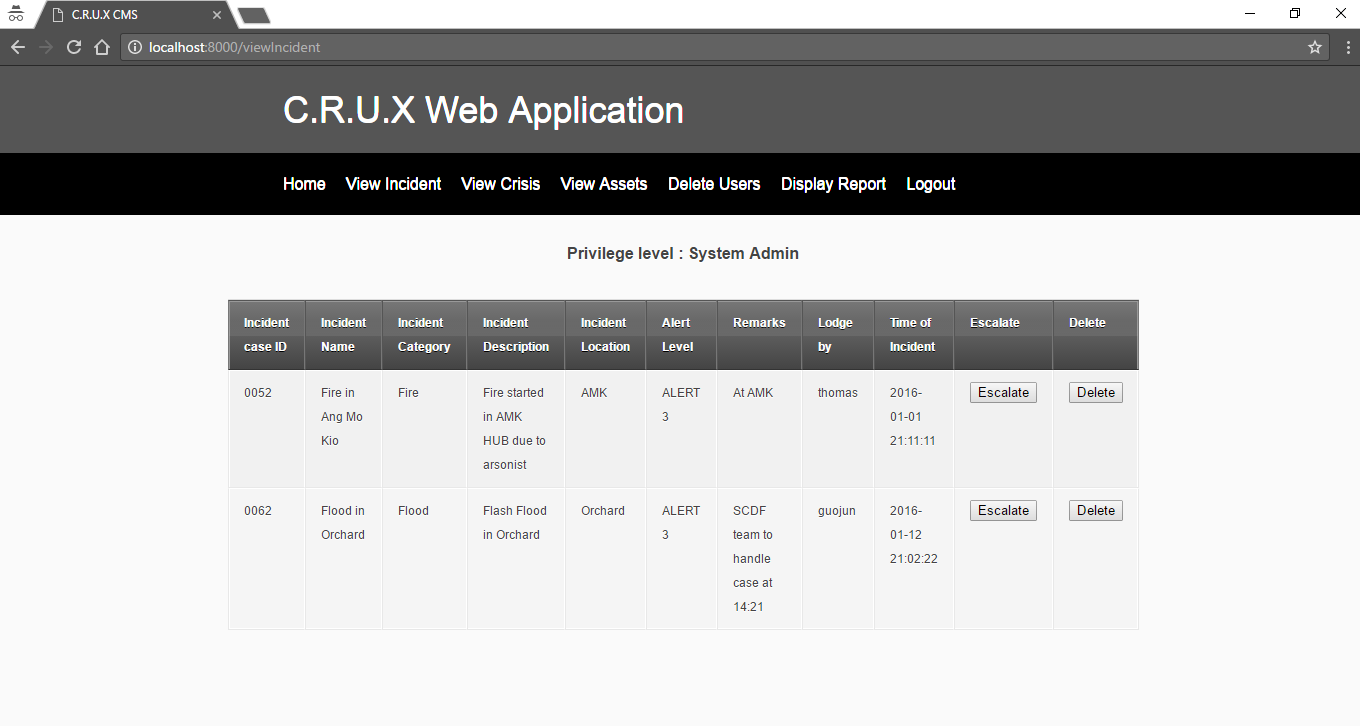
Call operator is able to log incident into the system when he receive a call or informed by the public.

**4.1.1.4 Edit Incident**

****

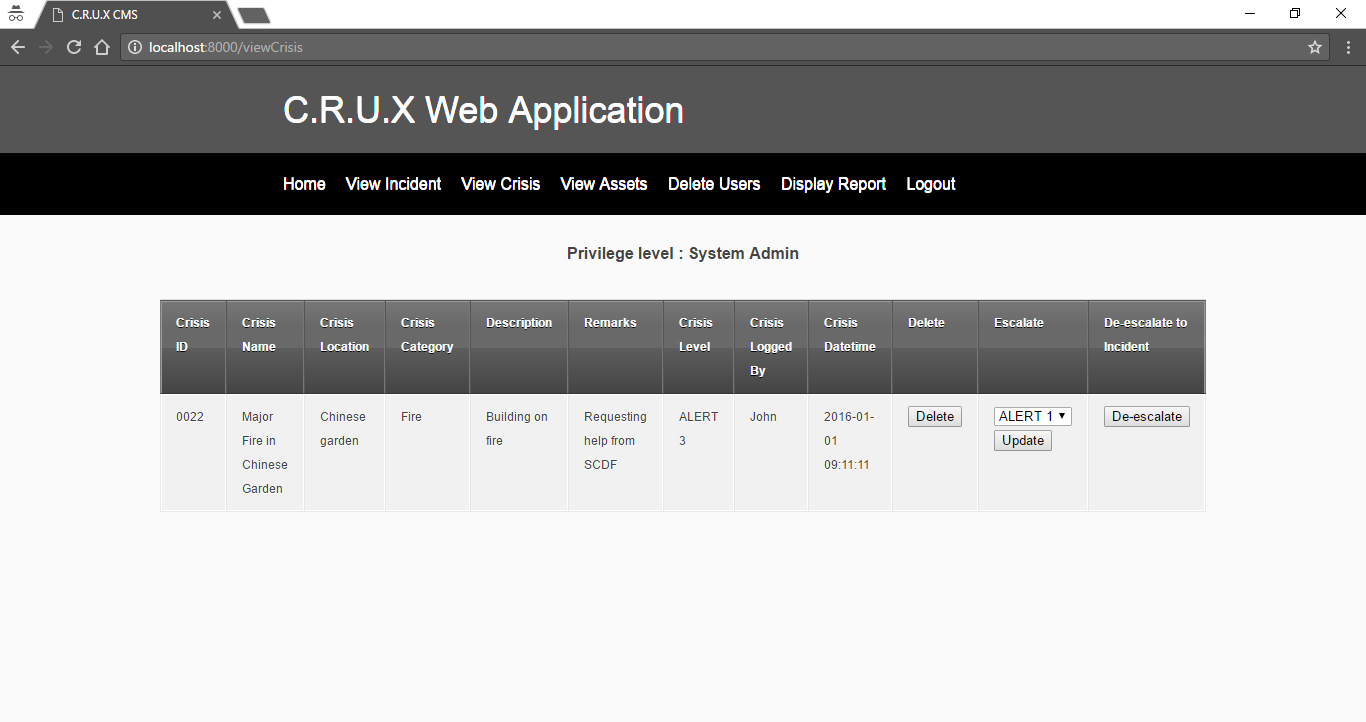
At any point of time where an incident needs to be updated, operator is able to update inside through this page.

**4.2.1.5 View Incident**



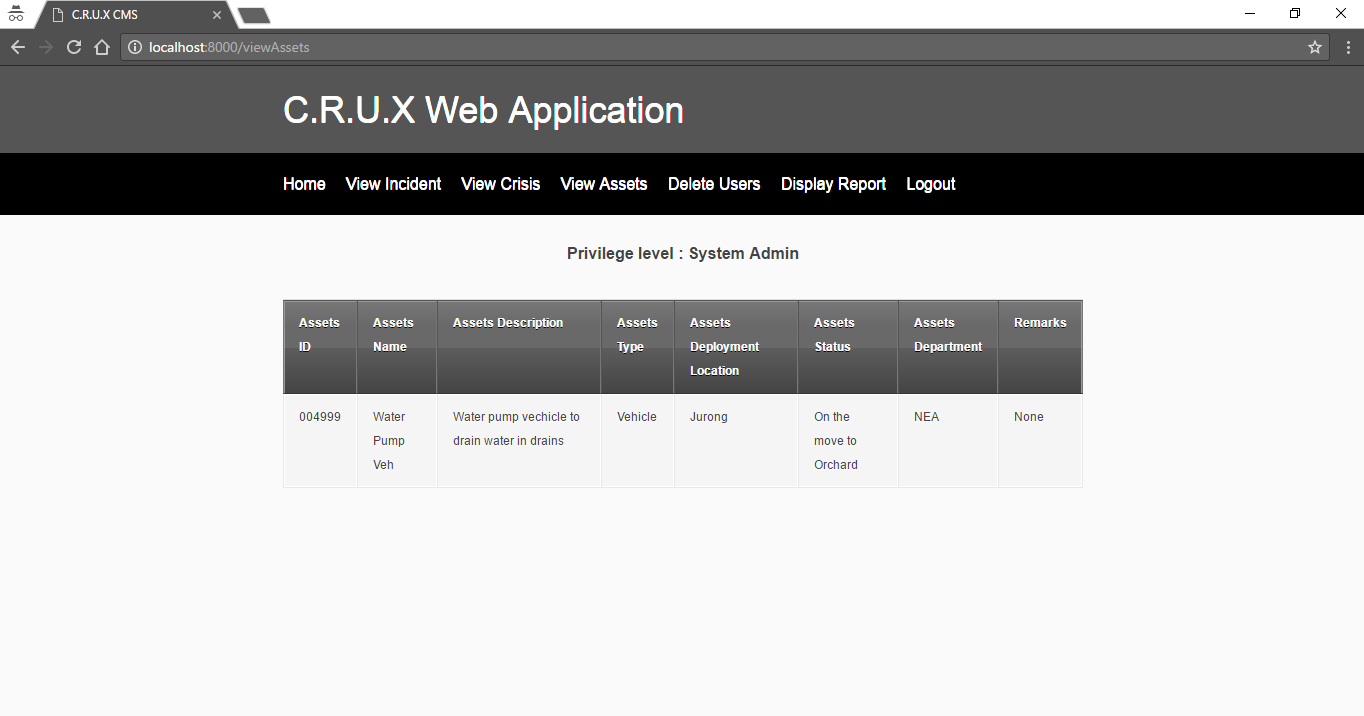
System admin is able to escalate incident to crisis at the view incident page if an incident becomes too serious. He can also remove the incident from the system if he deem it is too minor according to the policy.

**4.2.1.6 View Crisis**



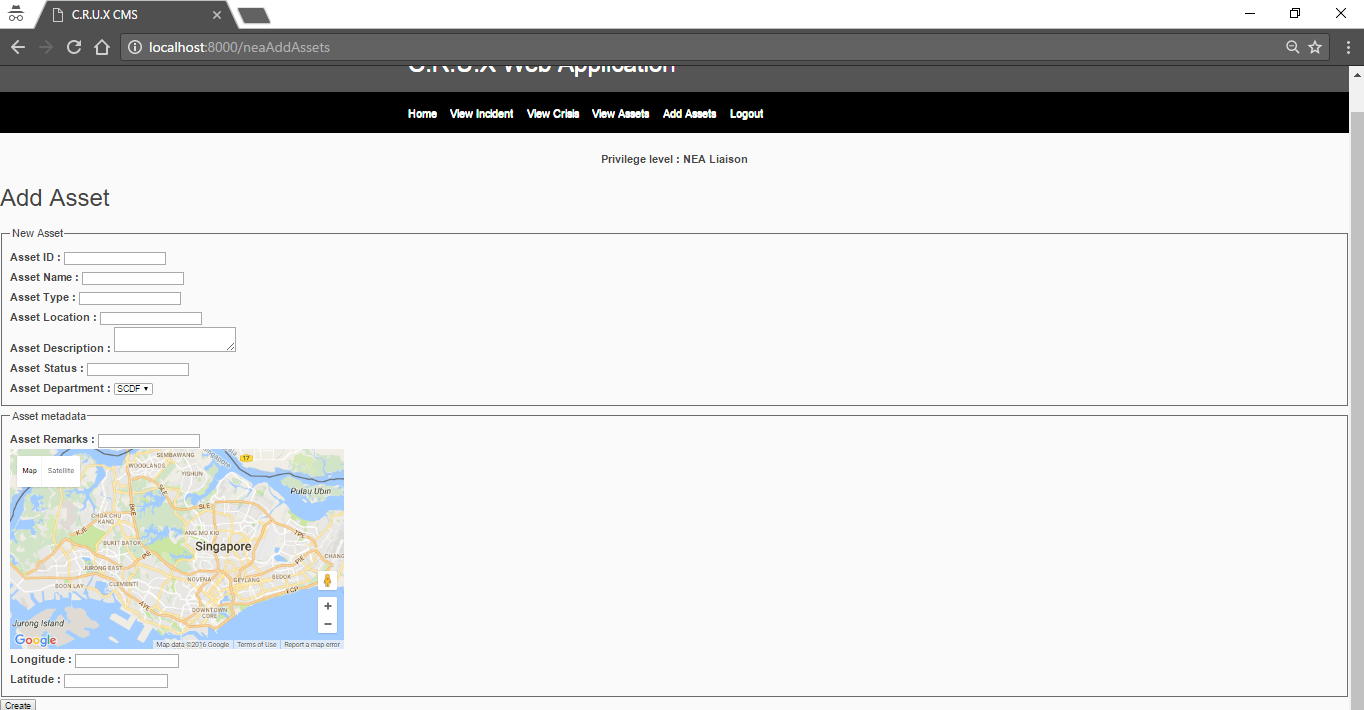
If the System admin choose to escalate an incident to a crisis, it will be shown on the crisis page. System admin is able to de-escalate the crisis to an incident anytime depending on what is stated in the policy.

**4.2.1.7 View Assets**

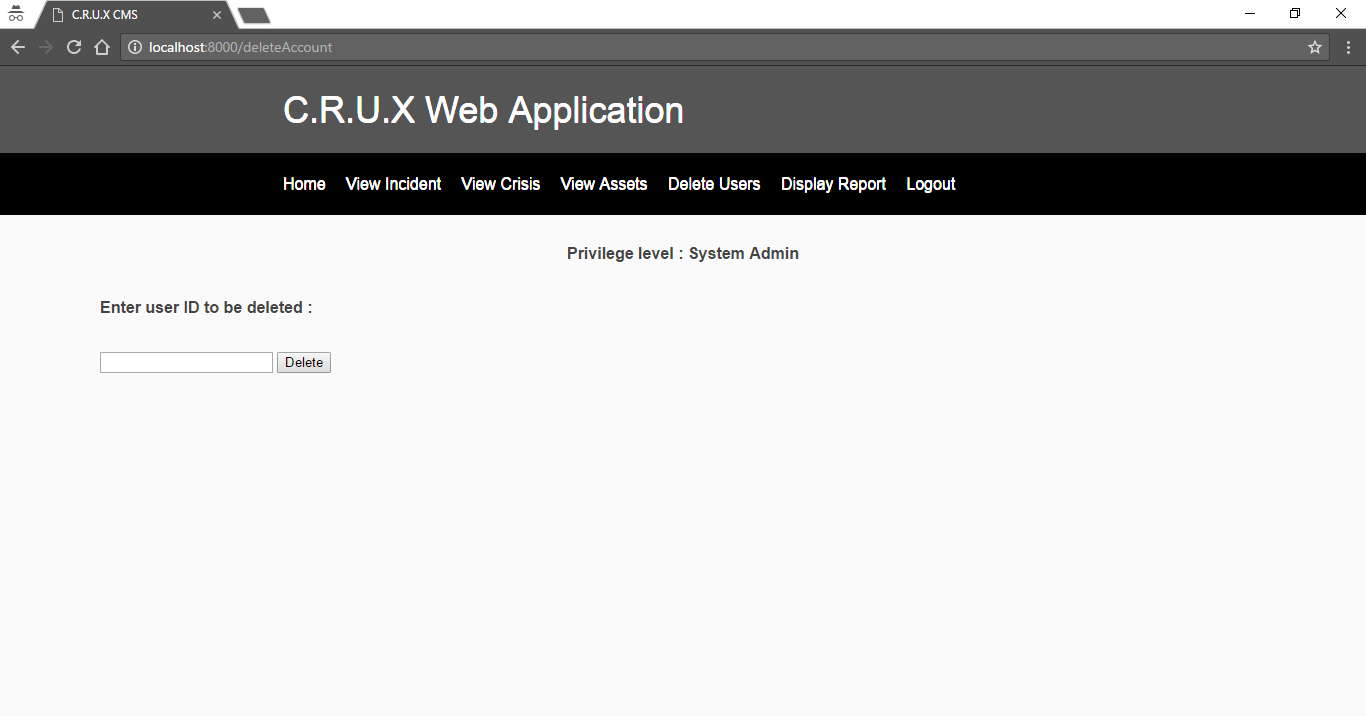


Assets will also be displayed in a page as shown above. This will allow commanders to easily keep track of assets deployed in the field for different incidents or crisis.

**4.2.1.8 Add Assets (SCDF/NEA/SPF/SAF)**

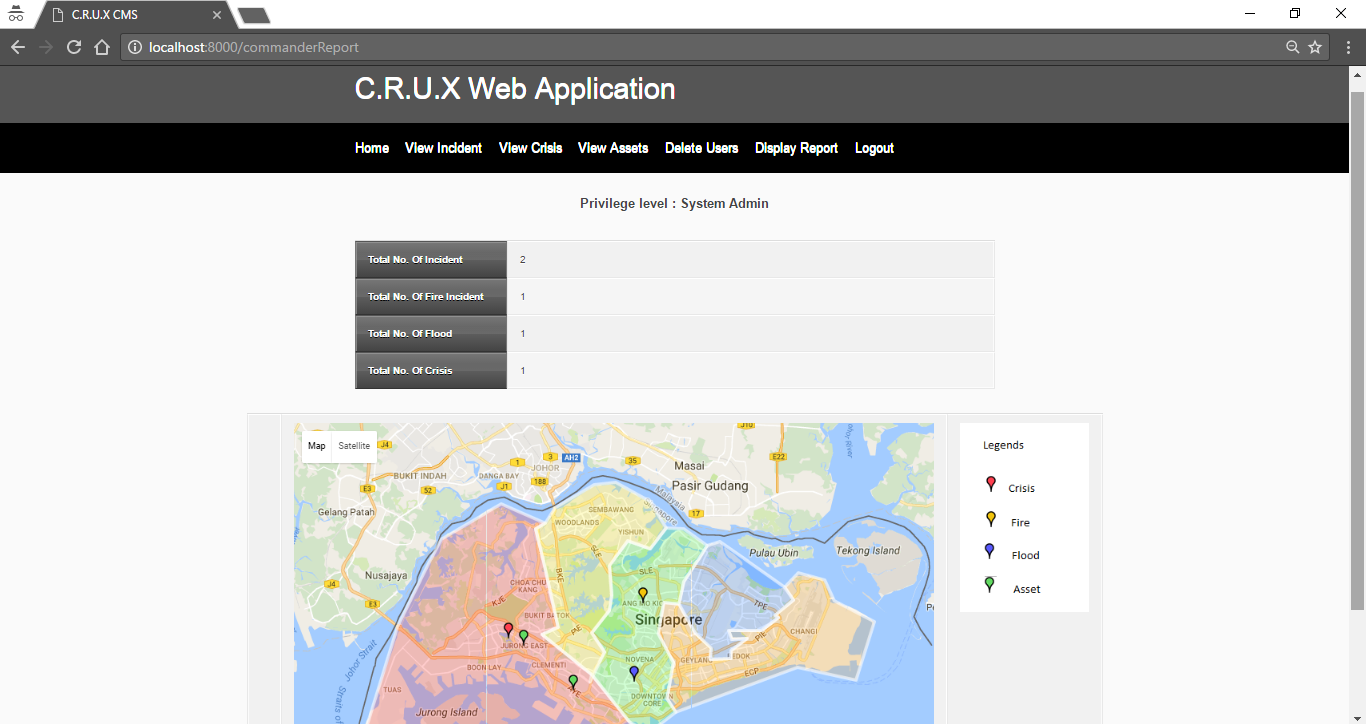


**4.2.1.9 Delete Users**



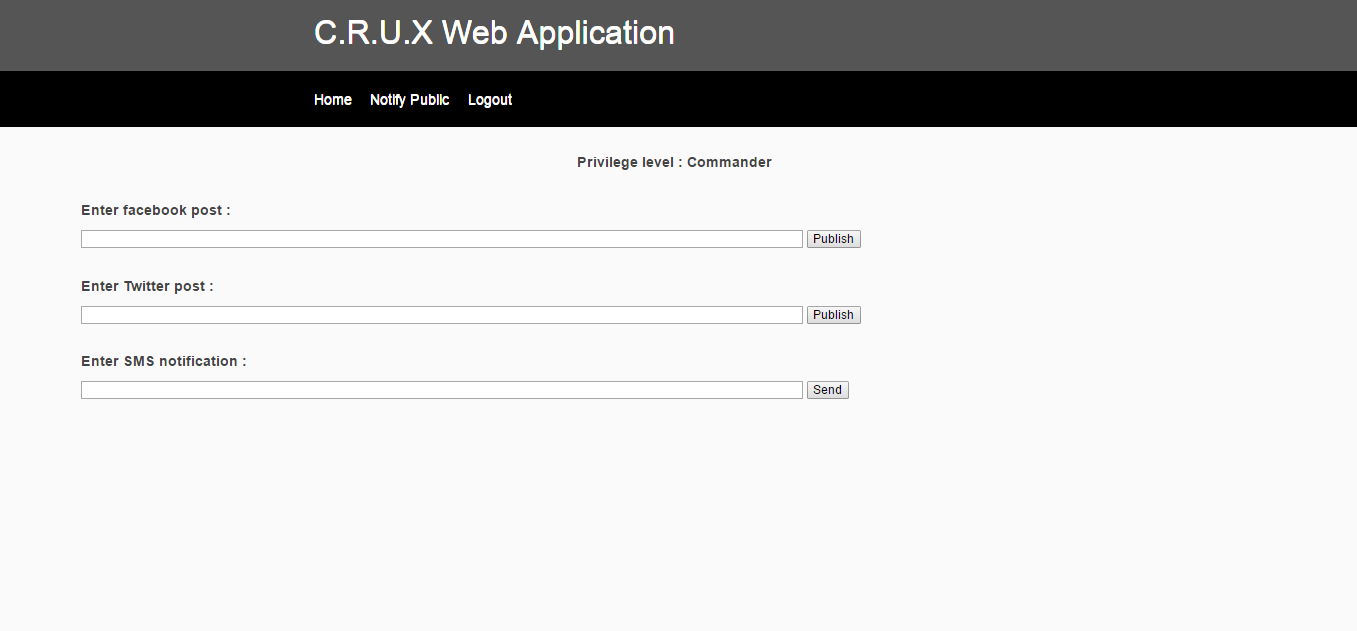
System Admin is also able to remove users from the system if the account is not in use anymore.

**4.2.1.10 Display Report**

****

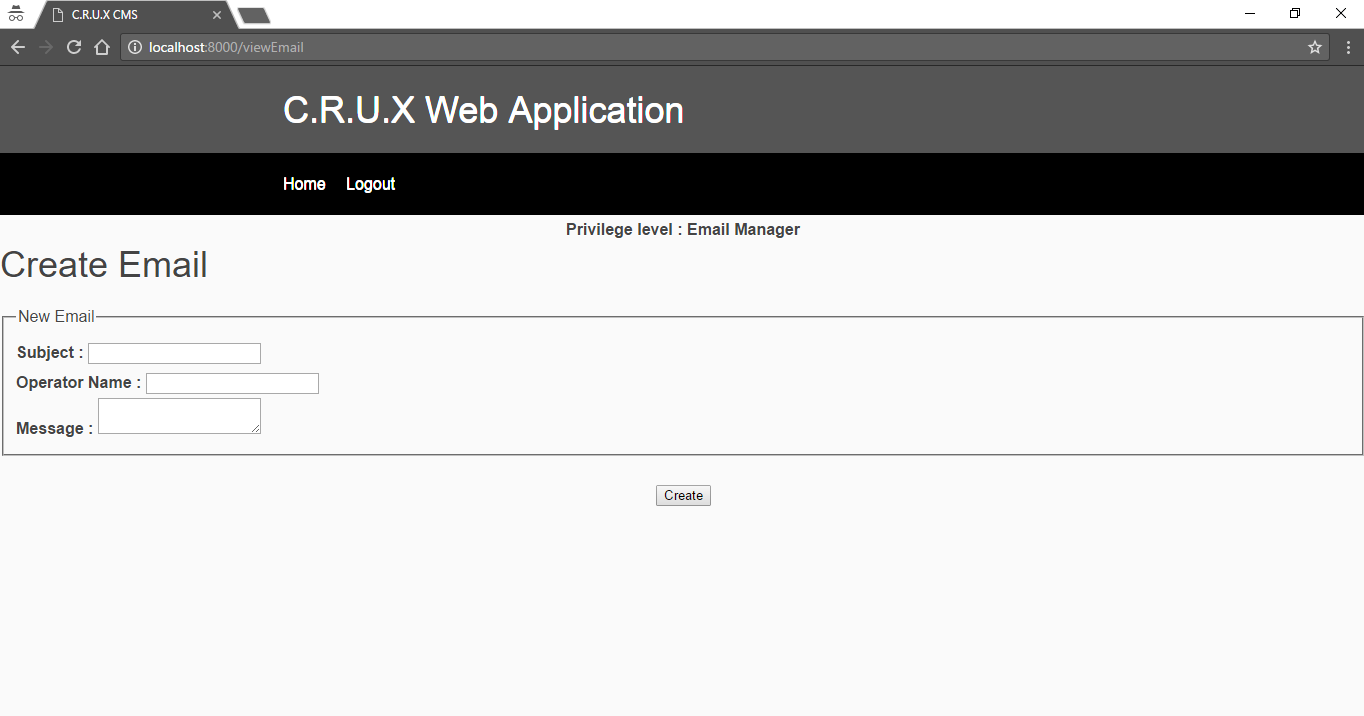
There will also be a report page where user is able to view the total number of incident/crisis happening currently.

**4.2.1.11 Notify Public**



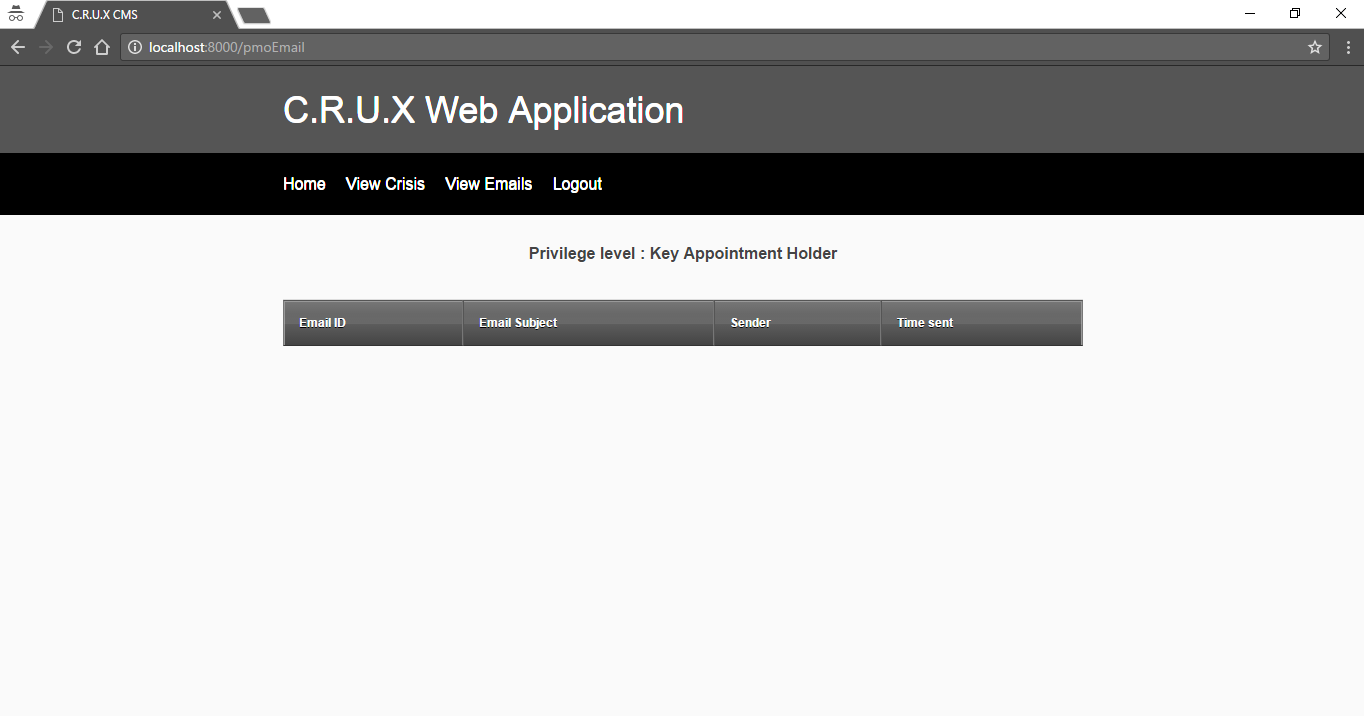
Public Liaison is able to notify public through twitter, Facebook or SMS through this page.

**4.2.1.12 Email PMO**

****

As PMO will need to review on incident and crisis periodically, an email will be crafted by the email manager and send to the key appointment holders to facilitate their decision making.

**4.1.1.13 View Email**



In this page, Key appointment holders will be able to view email reports sent giving them additional information about the incident and crisis.

**4.2.1.14 Users Scope**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Role**    **Scope** | **View**  **Home** | **Lodge Incident** | **View Incident** | **Delete Incident** | **Update**  **Incident** | **Escalate Incident** | **View Crisis** | **Delete Crisis** | **View Map** | **Send Email** | **View Email** | **Add Assets** | **Delete Assets** |
| System Admin | **✓** |  | **✓** | **✓** |  | **✓** | **✓** | **✓** | **✓** |  |  |  |  |
| Call Operator | **✓** | **✓** | **✓** |  | **✓** |  |  |  |  |  |  |  |  |
| PMO | **✓** |  |  |  |  |  | **✓** |  | **✓** |  | **✓** |  |  |
| SAF Commander | **✓** |  | **✓** |  |  |  | **✓** |  | **✓** |  |  | **✓** | **✓** |
| NEA Liaison | **✓** |  | **✓** |  |  |  | **✓** |  | **✓** |  |  | **✓** | **✓** |
| SCDF Commander | **✓** |  | **✓** |  |  |  | **✓** |  | **✓** |  |  | **✓** | **✓** |
| SPF Commander | **✓** |  | **✓** |  |  |  | **✓** |  | **✓** |  |  | **✓** | **✓** |
| Email Liaison | **✓** |  | **✓** |  |  |  |  |  |  | **✓** |  |  |  |
| Public Liaison | **✓** |  | **✓** |  |  |  |  |  |  |  |  |  |  |

**Figure 4.** *Please note that figure does not show every possible operations or functions, but just the major ones.*

### 4.2.2 Hardware Interfaces

Users are highly recommended to have:

* Keyboard
* Mouse
* Monitor
* Desktop Computer
* At least a 56k modem for stable internet access

### Software Interfaces

Operating System

A functional operating system that has internet connection is required for C.R.U.X. However, a Window or Linux system is preferred.

Database

The system will be using **MySQL 5.7**, an open-source relational database management system mainly for managing incidents, crisis, assets information, as well as user accounts.

Web browser

The system will be able to run on any web browser such as Firefox, Google Chrome and Internet Explorer.

### Communication Interfaces

The CMS shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

The system will communicate with other APIs:

* Google Map API
* Weather API
* Twitter API
* Email API
* SMS API

# System Features

## 5.1 Functional Requirements

### 5.1.1 Login

1.1. The system shall prompt the user to key in his/her username and password.

1.2. The user must enter the username and password.

1.3. The system must validate the username and password entered.

1.3.1. In the event if the user had entered an invalid or empty username and/or password:

1.3.1.1. The system shall display error message: “Invalid username/password entered! Please try again.”

1.4. The user must select on the “Login” button.

### 5.1.2 Create/Log New Incident

2.1. Upon receiving a call or informed by the public:

2.1.1. The user (Operator) must input the Case ID.

2.1.2. The user (Operator) must input the Incident Name.

2.1.3. The user (Operator) must input the Operator Name.

2.1.4. The user (Operator) must input the Incident Location.

2.1.5. The user (Operator) must input the Incident Timing.

2.1.6. The user (Operator) must input the Incident Description.

2.1.7. The user (Operator) must select the Alert Level with a default value of 1 from the drop down list.

2.1.8. The user (Operator) must select a category for the incident either as Fire or Flood.

2.1.9. The user (Operator) may or may not input remarks with regards to the incident.

2.1.10. The user (Operator) shall place a marker on the map with respective to the location of the incident, reflecting the Longitude and Latitude on the textboxes accordingly.

2.1.11. The user (Operator) shall click on ‘Create’ upon checking all the required fields above, making sure there are no blank fields except the ‘Remarks’ field being optional.

2.1.11.1. In the event if there are invalid and/or blank field(s), the system will validate an error message and the user (Operator) will need to ensure there are no blank and/or invalid fields except the ‘Remarks’ field.

### 5.1.3 Edit Incident

3.1. The user (Operator) shall select on the ‘Update’ button with respect to the Incident Case ID of the Incident Name that he/she wishes to edit.

3.2. The user (Operator) will be directed to the same page as ‘Create New Incident’ with exactly the same fields, just that the fields now are all modifiable (including the map that can be updated with a new marker).

3.2.1. In the event upon clicking the ‘Update’ button after the user (Operator) is done with all the modification:

3.2.1.1. If the user (Operator) had keyed in invalid and/or blank field(s), the system will validate an error message and the user (Operator) will need to ensure there are no blank and/or invalid fields except the ‘Remarks’ field.

3.3. The system will successfully update the new modified fields.

### 5.1.4 View Incident

4.1. The user (System Admin) must select on the “View Incident” tab to view and monitor the ongoing incidents.

4.2. The system must redirect to the incident page and list the existing incident(s) that was/were created by the Operator.

4.3. The user (System Admin) shall escalate the selected existing incident to crisis if an incident becomes too serious.

4.3.1. In the event if the user has selected “Escalate” for a specific incident, he/she will be directed to the crisis page with the incident escalating to a crisis.

### 5.1.5 View Crisis

5.1. The user (System Admin) must select on the “View Crisis” tab to view and monitor the ongoing crisis.

5.2. The user (System Admin) shall de-escalate the crisis to an incident anytime (when situation is better) based on the Policy definition shown at the Appendix G.

5.3. The user (System Admin) shall delete the crisis where it deems fit.

5.4. The user (System Admin) must click on the drop down list to select an Alert Level to suit the crisis at that point in time, followed by clicking on the “Update” button.

### 5.1.6 Delete Incident

6.1. The user (System Admin) must select on the “View Incident” tab.

6.2. The system must redirect to the incident page and list the existing incident(s) that was/were created by the Operator.

6.3. The user (System Admin) shall select on “Delete Incident” when the incident had successfully died down (either formerly from a crisis or as an incident itself).

### 5.1.7 View Map

7.1. The user (System Admin and/or Commander) must select on the “Home” tab.

7.2. The user (System Admin and/or Commander) shall be able to view the Singapore map with markers that indicate incidents, crisis, and assets deployed around Singapore.

7.2.1. The user (System Admin and/or Commander) must be able to see the different colours of the markers of the various existing incident(s)/crises with the legend symbolizing different meaning of them (at the right side of the page).

7.3. The user(s) (Commanders) shall be able to chat among each other using the chat provided at the left side of the page regarding the status of the crisis.

### 5.1.8 View Assets

8.1. The user (System Admin and/or Commander) shall be able to view the assets by clicking on the “View Assets” tab.

8.2. The system shall list the details of the assets pertaining to the incident that was being logged earlier.

8.2.1. The details of the assets are as such:

8.2.1.1. Assets ID, Assets Name, Assets Description, Assets Type, Assets Deployment Location, Assets Status, Assets Department, and Remarks.

8.3. The user (System Admin and/or Commander) shall be able to keep track of assets deployed in the field for different incidents and/or crises.

### 5.1.9 Add Assets (SCDF/NEA/SPF/SAF)

9.1 The system must validate all user input.

9.2 All fields must not be left empty.

9.3 The system must display map of Singapore on the page.

9.4 The system must display longitude and latitude after a marker is added onto the map.

### 5.1.10 Delete Users

10.1 The system must validate user input.

10.2 The system must delete user from the database if user input is valid.

### 5.1.11 Display Report

11.1 The system must display total number of incidents.

11.2 The system must display total number of fire incidents and total number of flood incidents.

11.3 The system must display total number of crisis.

11.4 The system must display the map of Singapore on the page.

11.5 The map must display the location of crisis, incident and asset on the map through the use of marker.

11.6 The markers on the map must be correctly coloured coded (Red for crisis, Yellow for fire incident, Blue for flood incident and green for asset).

### 5.1.12 Notify Public

12.1 The system must allow user to select the platform (Facebook, Twitter, SMS) to send notification.

12.2 When SMS notification is selected, the system must send a SMS to all public members subscribed to the service.

12.3 When Twitter notification is selected, the system must post a tweet to the Twitter page of application.

12.4 The system must send the email to only relevant key appointment holder (e.g. only PMO and key appointment holder of SCDF will receive an email for fire related crisis).

### 5.1.13 Email PMO

13.1 The system must have the email address of the PMO and key appointment holder.

13.2 The system must automatically create a latest status report summarizing key indicators and trend after creation of an incident.

13.3 The system must be able to send out email automatically after the status report is created.

13.4 The system must send the email to only relevant key appointment holder (e.g. only PMO and key appointment holder of SCDF will receive an email for fire related crisis).

### 5.1.14 View Email

14.1 The system must store all emails sent to each user.

14.2 The system must allow user to view the content of each email.

## 5.2 Software Requirements

The system will be hosted as a web application and it will be a private application only catered for selected users. The system must be

* Compatible with chrome v48 and above
* Internet explorer v11 and above
* Firefox v44 and above

## 5.3 User Interface Requirements

* User interface must be consistent throughout different pages and browsers.
* Logout button must always be available in the sidebar
* Color Theme and layout of user interface must not use more than 4 colors.
* User must be able to navigate to different pages according to his scope using the sidebar
* Error message must be displayed
* Confirmation message must be displayed

# Other Non-Functional Requirements

## 6.1 Performance Requirements

6.1.1 SMS notification sent must reach receiver within 5seconds.

6.1.2 Map updates on asset, incident and crisis must not take more than 5seconds.

6.1.3 Time taken for social media to be posted must be within 5 seconds.

6.1.4 Create, update, delete and read operation must not exceed 3seconds.

6.1.5 Assets must be displayed on the map less than 3second once it has updated its status and location.

6.1.6. Email sent to PMO must be timely and accurate, with less than 30seconds off the exact time.

## 6.2 Security Requirements

6.2.1 Password must be encrypted in the database.

6.2.2 Password field must be hidden.

6.2.3 Accounts are created by system admin hence user is not able to create account from the web application page itself for security purposes.

6.2.4 Each user can only able to access pages based on their access privilege.

6.2.5 Each user can only perform their set of allowed operations on the system.

## 6.3 Extensibility Requirements

6.3.1 System must be able to support new database with the use of Data Access Object that supports configuration for MySQL and SQLite.

**6.4 Maintainability Requirements**

6.4.1 Changes and fixes of codes to one component will not affect another component (View not affecting another View & Controller not affecting Controller)

6.4.2 Codes must be arranged neatly into directories based on whether it is a Model, View and Controller.

## 6.4 Software Quality Attributes

**6.4.1 Availability and Reliability**

Since the ‘problems’ can be of many types, different technologies work in tandem to achieve availability for the overall system.

As C.R.U.X is a life critical system, we aim to

* Allow users to log on to the system anytime of the day (24/7/) uptime
* Allow a minimum of 10 users to use the system concurrently
* Have a stable system with a success rate of 99% uptime rate ( high availability)

**6.4.2 Flexibility**

Flexibility is important to C.R.U.X because

* The system must be able to support new changes in crisis category(e.g. terrorism, disaster outbreak)
* The system must be able to support any new additions to supporting government agencies (e.g. volunteer corps)
* The system must be able to support new functions if needed (e.g conference call, online chat)

**6.4.3 Portability**

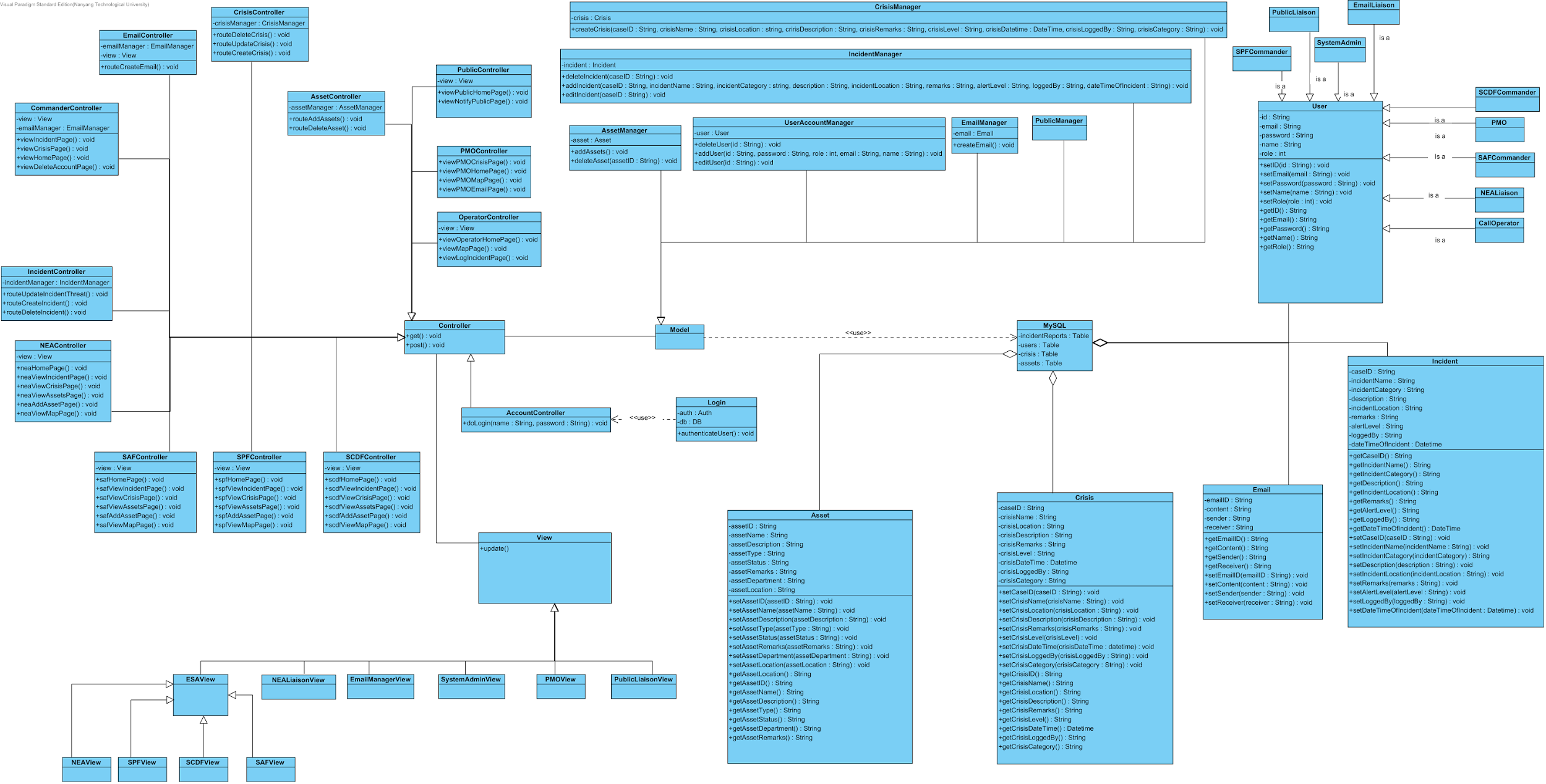
C.R.U.X must be able to function consistently on multiply browser platform such as safari, google chrome and internet explorer.

# Appendix A: Glossary

|  |  |
| --- | --- |
| **Name** | **Terms** |
| Government Agencies (GA) | Refers to the government sectors involved in the crisis. For example, Ministry of Health, Ministry of Education, etc |
| Emergency Service Agencies (ESA) | Refers to the different emergency units that are involved in the incident |
| Category of Incident | Refers to the different types of incident that has happened   * Fire * Flood |
| Head of Staff (HoS) | Refers to the person in-charge of the whole incident and who has the authority to delete or amend the crisis |
| Archive Incident | Archive Incident for future record purpose |
| Email Liaison | Officer in charge to send email to PMO periodically |
| SAF Commander | Officer in charge to handle assets and situation from SAF side |
| SCDF Commander | Officer in charge to handle assets and situation from SCDF side |
| SPF Commander | Officer in charge to handle assets and situation from SPF side |
| NEA Liaison | Officer in charge of handling assets and situation from the NEA side |
| Email API | Refers to application programming interface that helps dispatch the email to the people associate to the incident |
| Weather API | Refers to application programming interface that helps the users view the current weather condition |
| Login | Refers to the login CMS before viewing /using features provided by the system. |
| Update Crisis | Refers to updating of the incident so that the system will be in synchronization with the latest news received. |
| Delete Crisis | Refers to the choice of crisis to be deleted, the existing Crisis regardless the reasons. |
| Archived Crisis | Refers to the choice of crisis to be archived from existing incident instead of deleting the crisis |
| Create Crisis | Refers to the creation of a new incident in the CMS. |
| Dispatch information | Refers to the dispatchment of information by the user to the people/ministries involved in the crisis |
| Monitor incident | Refers to main page of the CMS showing a list of incident logged by ESA and other sources |
| Track asset | Refers to the accessing of the amount of asset activated/used for a particular incident |
| View Weather | Refers to monitoring of the current weather situation through weather API |
| View Incident/Crisis | Refers to the viewing summary of the incident/ crisis |
| Upload Social Media Update | Refers to the different platforms (Twitter,SMS) which the PR Team chooses to dispatch the information to the public |

# Appendix B: Analysis Models

## 8.1 Class Diagram

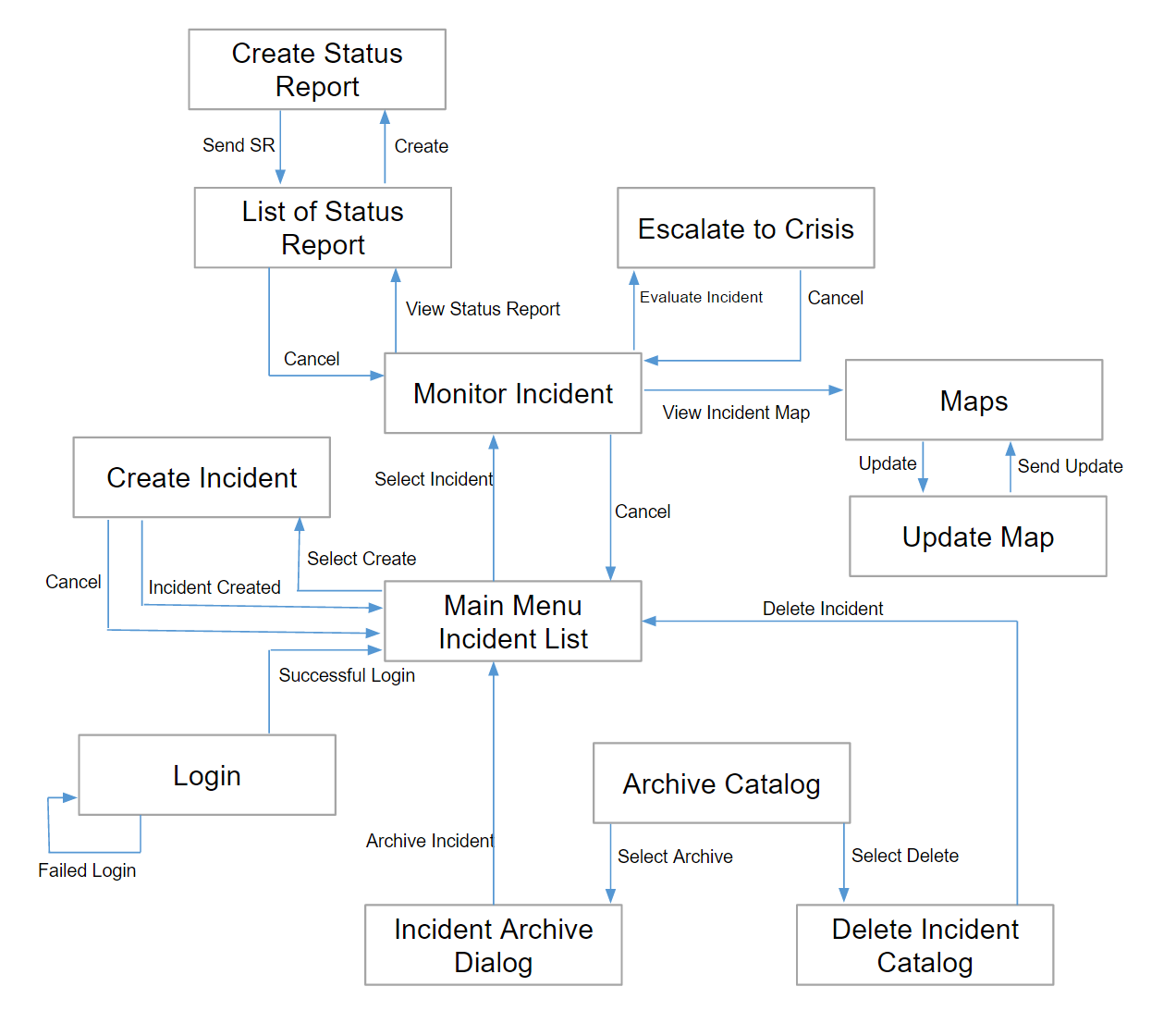


## 8.2 Context Diagram

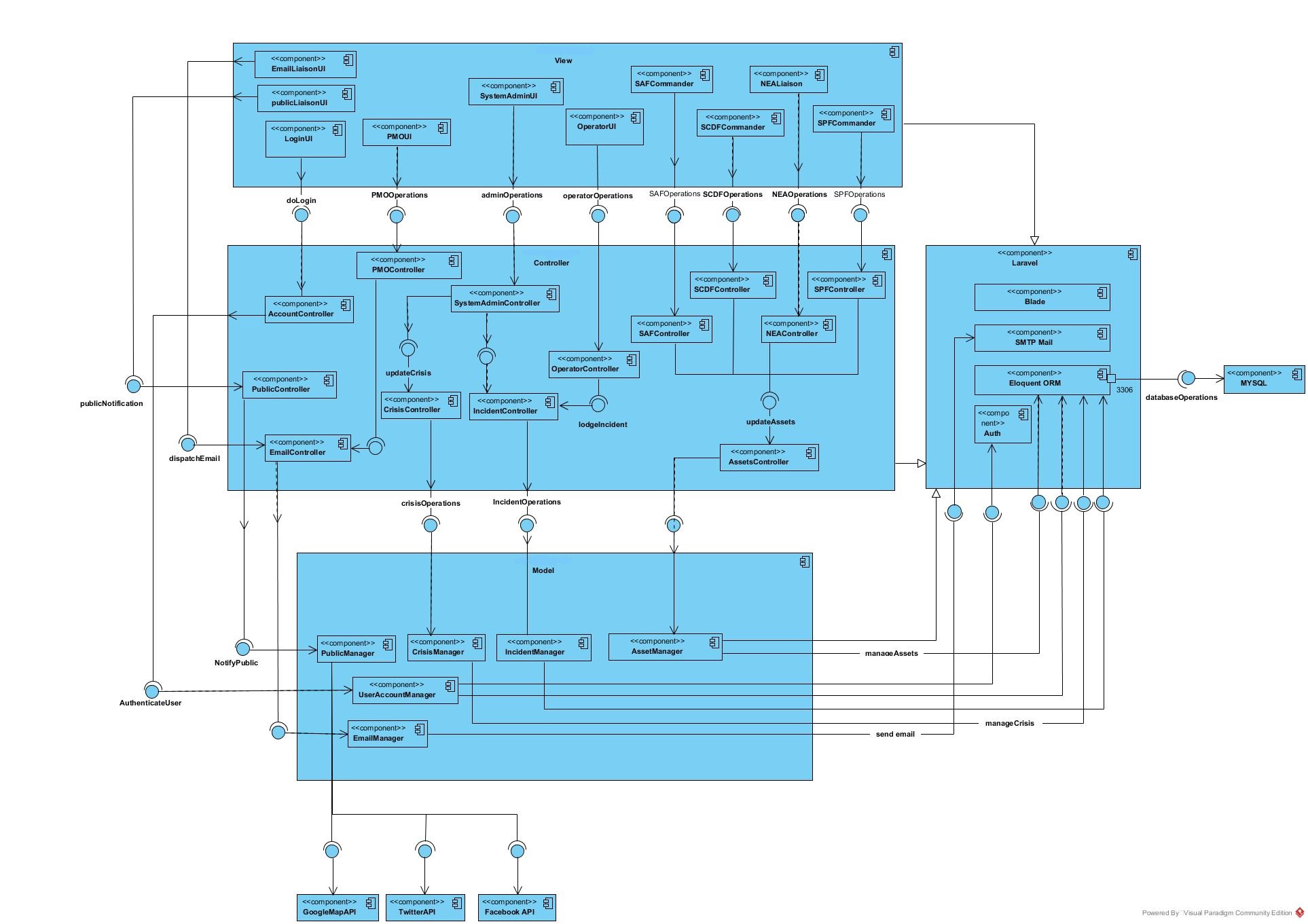
## 8.3 Data Flow Diagram

## 

## 8.4 Dialog Map



## 8.5 Architecture Diagram



C.R.U.X uses Laravel, a web PHP framework that follows the Model-View-Controller (MVC) architecture style. Laravel framework has a very straightforward file directory system that has individual folder for each Model, View, and Controller. Developers will be able to easily find/identify the folders of a particular component they are working on.

Apart from the basic MVC structure Laravel provides, it also introduces several basic web features that our system taps on:

1. Blade - template for View creation

2. Eloquent ORM - a Data Access Object(DAO) that interacts with the application database

3. Auth - Basic authentication structure for web application. Works in hand with Eloquent ORM for data accessing into MySQL database to verify if user exist and whether their credential is valid

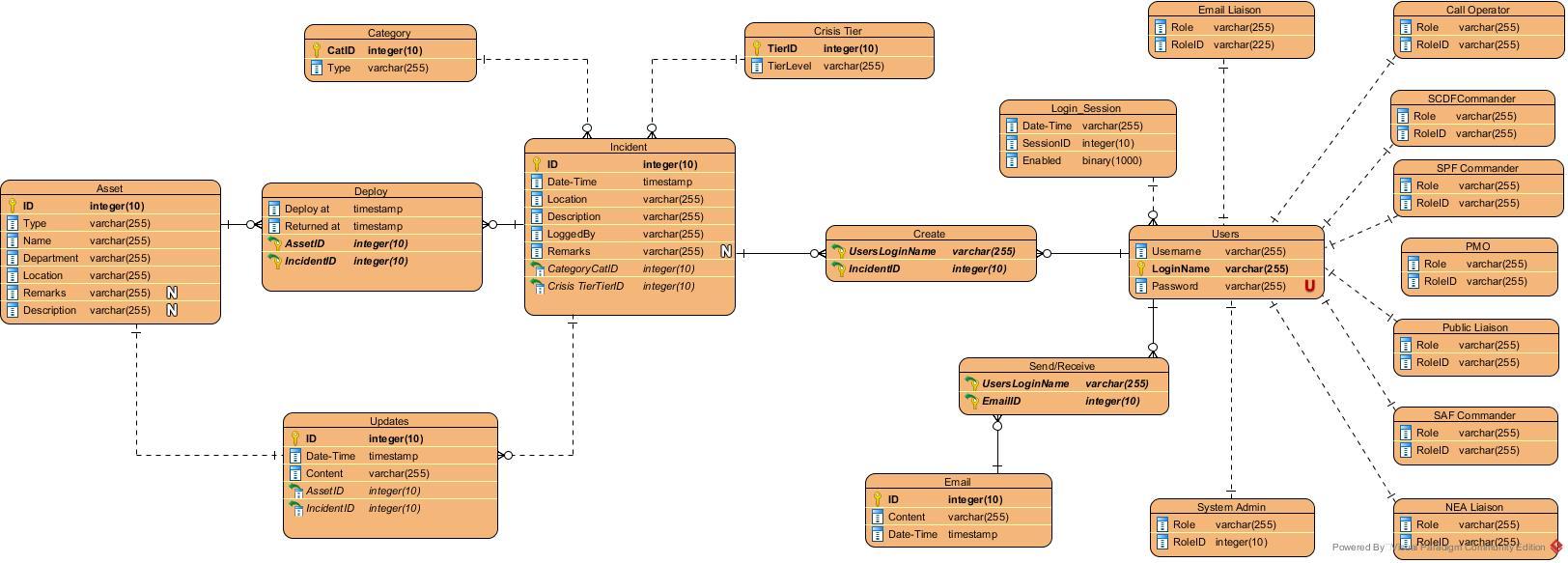
Leveraging the features of Laravel, C.R.U.X is developed in the following manner:

- User Interface (UI) of individual roles are created inside the "View" folder, for each role having different sets of operation.

- UI will receive inputs and pass it to it individual controllers.

- Controllers will process the inputs, and reaching out to the database if required by going through the managers housed inside the "Model" folder

**8.6 Entity -Relational(ER) Diagram**



# Appendix C: Policy Definition

This appendix will define the various THREATCON level on the situation of natural disaster crisis. THREATCON is a measure of action and readiness to be carried out in a given situation. These actions will be used by various emergencies task forces such as Singapore Civil Defence Force, Singapore Police Force, Singapore Armed Forces and Public Utility Board.

Situation A - Occurrence of flash flood.

Situation B - Major fire outbreak.

**THREATCON 1**

THREATCON 1 will be declared if either one of the situation are stated below occurred at any time.

Situation A - Heavy downpour of continuous rain for 6 hour and water level is 0.5m tall.

Situation B - Fire outbreak and SCDF is trying to control for 2 hour.

Key decision makers will carry out if THREATCON 1 is activated:

Situation A - Notify public to remain indoor until rain subside.

Situation B - SCDF to activate Fire Fighting Force Bravo (3x red rhino) to on high alert and standby.

**THREATCON 2**

THREATCON 2 will be declared if either one of the situation are stated below occurred at any time.

Situation A - Reports of water level above 1.25m tall.

Situation B - Fire spread to 1 hectare wide.

Key decision makers will carry out if THREATCON 2 is activated:

Situation A - SPF to activate Contingency Plan Alpha of closure of roads in affected area and display of flash flood warning at LED signboards using EMAS to inform motorists. Dispatch SPF Alpha team to aid public that are stranded outdoor due to flood.

Situation B - Notify public to avoid incident area and SAF to deploy Alpha team to provide immediate first aid alert.

**THREATCON 3**

THREATCON 3 will be declared if either one of the situation are stated below occurred at any time.

Situation A - water current running at speed of 25m/s

Situation B - Smog level reached above 150 PM2.5

Key decision makers will carry out if THREATCON 3 is activated:

Situation A - SCDF deploy Alpha Team to aid PUB in pumping flood water out of any flooded area and Bravo Team (10x5 pax Lifesaver Boat) to assist stranded civilians. SAF to deploy immediate first aid station.

Situation B - SCDF deploy Charlie Team (10x Red Rhino) and SAF to deploy fire fighting helicopter to mitigate further spread of fire. SPF to deploy alpha team and issue N95 masks to civilians within 2km of incident area.

# 10. Appendix D: Meeting Minutes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Lab Session | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 24 August 2016, 10:30 am – 12:30 pm, SW Lab 3 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang  · Goh Ka Hian | | |  | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Selecting type of natural disaster to focus on. | Discussion on type of natural disaster and its policy definition. | | Taken by: All  Deadline: 07 September 2016 |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 31 August 2016, 4:30 – 3:00 pm, NTU Business Library Pod 3 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Goh Ka Hian  · Lim Zi Yang | | |  | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Discussion on policy definition of natural/elemental related disaster | Finalised policy definition of natural/elemental related disaster. | | Taken by: All  Deadline: 07 September 2016 |
| **Task 2** | Discussion on use case diagram for CMS | Came out with use cases | | Taken by: All  Deadline: 07 September 2016 |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 1 September 2016, 4:30 – 8:30 pm, TR+18 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang  · Goh Ka Hian | | |  | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Use case description | Spilt use cases to do use case descriptions | | Taken by: Guo Jun, Ka Hian, Long Xiang  Deadline: 07 September 2016 |
| **Task2** | Drawing of dialog map | Came out with dialog map base on use case description | | Taken by: Hao Zhe and Zi Yang  Deadline: 07 September 2016 |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 02 September 2016, 11:30 am – 1:30 pm, LWN Library Pod 4 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Goh Ka Hian | | | · Lim Zi Yang | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Context diagram | Came out with context diagram | | Taken by: Genevieve  Deadline: 07 September 2016 |
| **Task2** | Data flow diagram | Came out with data flow diagram map from use case diagram | | Taken by: Thomas and Genevieve  Deadline: 07 September 2016 |
| **Task3** | Review of use case description and use case diagram | - | | Taken by: All |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 05 September 2016, 2:30 pm – 4:30 pm, LWN Library Pod 5 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang  · Goh Ka Hian | | |  | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Finalising and compiling of lab 1 deliverables. | - | | Taken by: All |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Lab Session | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 07 September 2016, 10:30 am – 12:30 pm, SW Lab 3 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang  · Goh Ka Hian | | |  | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Revising use case diagram after review by lab supervisor | Finalise use case diagram | | Taken by: All  Deadline: 07 September 2016 |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 14 September 2016, 1:30 – 4:30 pm, TR20 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang | | | · Goh Ka Hian | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Refining of policy definition. | Discussion held and job was tasked to Thomas to be completed. | | Taken by: Thomas |
| **Task 2** | Drawing of architecture diagram. | Discussion held and job was tasked to Jian Wei to be completed. | | Taken by: Jian Wei |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 20 September 2016, 1:30 – 4:30 pm, SMU | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang | | | · Goh Ka Hian | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Finalise policy definition | - | | Taken by: All |
| **Task 2** | Dialog map does not reflect how incident escalate to crisis. | Update dialog map | | Taken by: Zi Yang |
| **Task 3** | Incomplete architecture diagram | Discussion on what components to include into architecture diagram | | Taken by: All  Deadline: 21 September 2016 |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Lab Session | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 21 September 2016, 10:30 am – 12:30 pm, SW Lab 3 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang  · Goh Ka Hian | | |  | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Architecture diagram was lacking in components after review by lab supervisor. | Discussion to come out with all possible components. | | Taken by: All  Deadline: 21 September 2016 |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 05 October 2016, 4:30 pm – 7:00 pm, TR18 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang | | | · Sim Long Xiang  · Goh Ka Hian | |
| **Chaired by** | | | | Thomas Lim |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Framework selection (ASP.net, Laravel, CakePHP or Django) | After discussing and comparing the frameworks, Laravel was chosen. | | Taken by: All |
| **Task2** | Workload distribution | -Code Implementation: Jian Wei & Ka Hian  -Class Diagram:  Guo Jun & Long Xiang  -Sequence Diagram: Thomas & Zi Yang  -Candidate Architecture:  Hao Zhe  -ER Diagram:  Genevieve | | Taken by: All  Deadline: 12 October 2016 |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 11 October 2016, 3:30 pm – 9:00 pm, TR17 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang  · Goh Ka Hian | | |  | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Quality attributes | Discuss possible quality attributes which our system can support | | Taken by: All |
| **Task2** | Download of necessary software to start code implementation. | Xampp, Composer, MySQL installed in our laptop so that everyone can contribute to the code implementation. | | Taken by: All |
| **Task3** | Workload distribution | -Code Implementation: Jian Wei, Thomas, Long Xiang & Ka Hian  -Report Documentation:  Guo Jun & Long Xiang  -Sequence Diagram:  Zi Yang-Software Quality Attributes  Hao Zhe  -ER Diagram:  Genevieve | | Taken by: All |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Lab Session | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 12 October 2016, 10:30 am – 12:30 pm, SW Lab 3 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang  · Goh Ka Hian | | |  | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Review of class diagram, ER diagram, and sequence diagram. | A few changes were made to the diagrams after discussion. | | Taken by: All |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 13 October 2016, 4:00 pm – 6:00 pm, SCSE Lounge Pod 4 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang  · Goh Ka Hian | | |  | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | How to ensure project file is consistent across all team members (Dropbox, Source Tree, etc.) | Source Tree selected to share project among us. Updated project can be uploaded using Source Tree and other team mates can download latest version from the application. | | Taken by: All |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 18 October 2016, 3.30 pm – 6:00 pm, TR+17 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Genevieve Lam  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang  · Goh Ka Hian | | |  | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Check progress of application development | -Review implemented features.  Discuss and trying possible methods to implement remaining features. | | Taken by: All |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 4 November 2016, 11.00 pm – 1:30 pm, Software Project Lab 1 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang | | | · Genevieve Lam  · Goh Ka Hian | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Component diagram | Diagram edited to be more clear and concise | | Taken by: Long Xiang |
| **Task2** | Presentation slides | Preparation of slides | | Taken by: Long Xiang & Guo Jun |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 6 November 2016, 3.30 pm – 8:30 pm, Nanyang Polytechnic | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang  · Goh Ka Hian | | | · Genevieve Lam | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Presentation and demo dry run | - | | Taken by: All |
| **Task2** | Chat feature in application | Chat feature implemented to allow users to chat within the application | | Taken by: Ka Hian & Thomas |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | | | | |
| Routine Group Meeting | | | | |
| **Date, Time (duration) and Venue** | | | | |
| · 7 November 2016, 9.45 am – 2:00 pm, NTU Business Library Pod 6 | | | | |
| **Attendees** | | | **Non-Attendees** | |
| · Sim Long Xiang  · Thomas Lim  · Huang Jian Wei  · Yong Guo Jun  · Lim Hao Zhe  · Lim Zi Yang  · Goh Ka Hian  · Genevieve Lam | | |  | |
| **Chaired by** | | | | Sim Long Xiang |
| **Progress Updates** | | | | |
| **Task** | **Problem/Issue/Progress** | **Solution/Action** | | **Taken by & deadline** |
| **Task1** | Report documentation | - | | Taken by: Long Xiang |
| **Task2** | Presentation dry run | - | | Taken by: All |
|  |  |  |  |  |