Software Requirements Specification

for

Crisis Management System

**Version 1.0 approved**

**Prepared by Team RakuNine**

**2nd September 2018**

**Table of Contents**

[**Introduction**](#_4eocipbffq2z) **4**

[Purpose](#_3znysh7) 4

[Intended Audience and Reading Suggestions](#_tk6f0vrbbf1s) 4

[Product Scope](#_16bsgw5wyhv9)4

[**Overall Description**](#_yi0ksgr6mdfm) **4**

[Product Perspective](#_2s8eyo1) 4

[Product Functions](#_outtse3r0eqd) 5

[User Classes and Characteristics](#_igxyd212ejba) 6

[Operating Environment](#_gjo0ei76jzfu) 7

[Design and Implementation Constraints](#_ov875z1ti2gg) 7

[Assumptions and Dependencies](#_kc1egtm7u1ml)7

[**External Interface Requirements**](#_2t0qxuoqsfx) **8**

[User Interfaces](#_2jxsxqh) 8

[Software Interfaces](#_eo46jemtbkfz) 10

[Communications Interfaces](#_wbcxkzlx4xik)11

[**System Features**](#_fq6h3cmxdoyq) **11**

[Provide Real-time Crises Information on Map](#_7oliwsw92vq2) 11

[Generate Status Report for Prime Minister](#_vk0gpabsbspr) 11

[Call Center Update](#_8fbbne77jaqx) 12

[Update Real-Time Data with API](#_2k2x4xi4wl59) 13

[Update Social Media](#_dn7pg71f7imw) 14

[Dispatch Command with SMS](#_1i7jdykm2kn9)14

[**Other Nonfunctional Requirements**](#_xm9stzm9hm1x) **15**

[Security](#_unxc31wvtlwn) 15

[Flexibility](#_pmhjz1krblo5)15

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This document exists to describe the requirements and specifications of the Crisis Management System (CMS). This is version 1.1.1 of the SRS and will describe the entire CMS.

## Intended Audience and Reading Suggestions

This document is intended for users, testers and system engineers responsible for maintaining the system. This documents contains the overall description of the Crisis Management System (CMS), requirements of the CMS and features of the CMS.

## Product Scope

The Crisis Management System (CMS) is developed with the aim of being a seamless collaboration between government agencies in times of emergency. On a graphical web interface, the CMS will provide real-time status updates on a map of Singapore, integrated with weather conditions, dengue hotspot, haze information and etc. The CMS will provide useful information such as location of Civil Defence shelters and updates of emergency situations like terrorist activities or accidents within crowded, confined areas to the public through social media periodically. The Prime Minister’s Office will also receive a status report summarizing key indicators and trends over email every 30 minutes.

When a member of the public reports an incident, the operators of the call centres will input it into the CMS, requests will be updated on the map and dispatched using SMS to relevant agencies.

Thus, key decision makers can monitor the crisis situation through the CMS and allowing the CMS to serve as a command-and-control platform to respond quickly to needs and dangerous conditions, such as gas leaks, hazardous air condition and fires. .

# Overall Description

## Product Perspective

Crisis Management System is a stand-alone system which monitors the real-time condition of dengue fever, haze and national security status in Singapore. This system comprises of two user interfaces, one for general public and one for call center, and the main server which is gathering data from api and call centers’ data, evaluate and filter them to generate status report and to detect emergency situation.

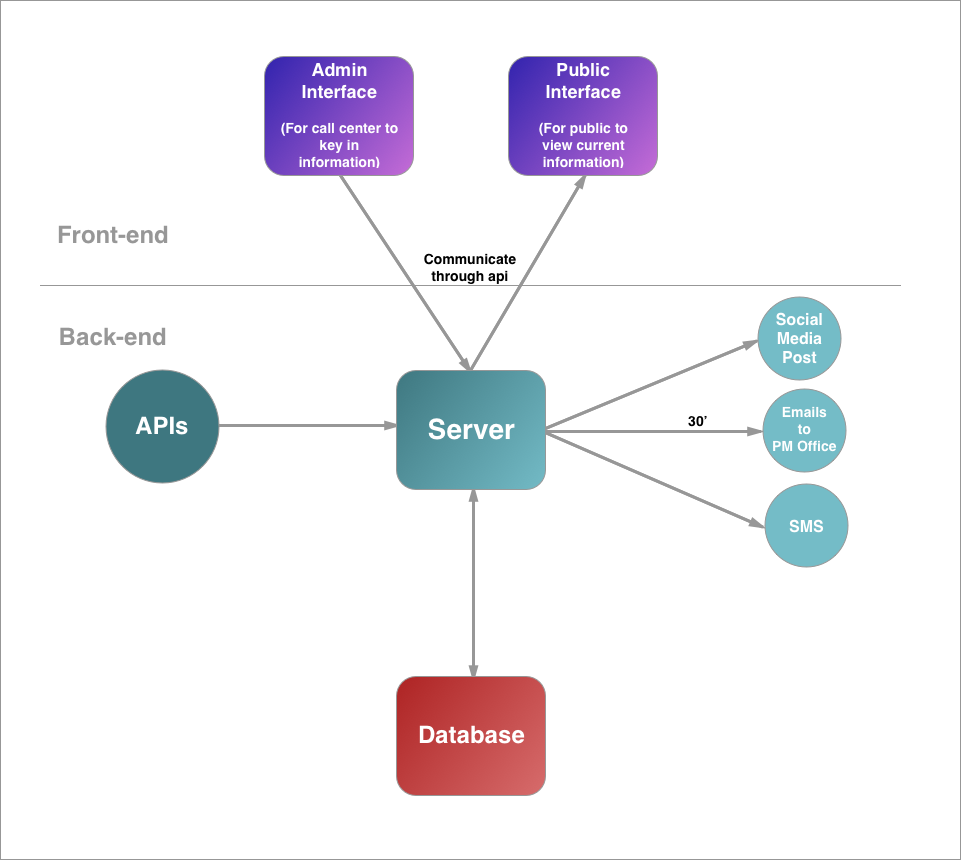


Fig: Front-end System and Back-end System

## Product Functions

The following functions are the major functions of our Crisis Management System (CMS) relating to different users-

**Public User’s functions**

1. Watch up-to-date data displaying on the Map

**Call Center’s function**

1. Update up-to-date information to the System’s database

**Government & Third party API’s function**

1. Update real-time information to the System’s database

**System’s functions**

1. Generate and send statistic report to Prime Minister’s Office
2. Send Dispatch Command to Relevant Government Agencies
3. Post up-to-date news on social media in case of emergency

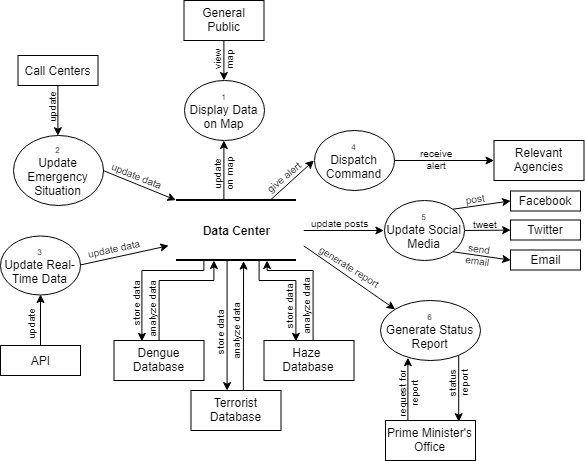


Fig: Data Flow Diagram

## User Classes and Characteristics

User Class - General Public

Frequency of Use - One to Many times depending on the situation

Technical Expertise - Novice

Security or Privilege levels - Read permission on system data and read, write permission for their user’s data

User Class - Call Centers

Frequency of Use - Many times

Technical Expertise - Trained Agents

Security or Privilege levels - Read,Write permission on system data

User Class - Relevant Agencies

Frequency of Use - One to Many times depending on the situation

Technical Expertise - Novice

Security or Privilege levels - Read permission on system data

User Class - Prime Minister’s Office

Frequency of Use - One to Many times

Technical Expertise - Novice

Security or Privilege levels - Read permission on system data

## Operating Environment

**Suitable Browsers**

1. Chrome
2. FireFox
3. Safari
4. Internet Explorer 11

**Suitable Hardware Platforms**

1. Any desktops, laptops or mobile devices which has access to internet

## Design and Implementation Constraints

**API Request**

Users need to refresh the browsers to update the data displaying on the interface. This is because the browsers only make the new API request to the server only when they are refreshed. This creates a big constraint on our real-time system.

**Database Storage**

The system will not just directly display data from API, but instead, it will store them in its own database first to filter and analyze them. A lot of storage database will be needed even though the system will wipes out the old data it has used.

**Display of Data on the Map**

The system will get the data from database and display it on the map. The interpretation of data to the level which map api understand consume processing power of the resources. So there will be trade off between real-time map performance and precision of data display.

## Assumptions and Dependencies

**Third-Party APIs**

This System depends on third-party apis from government and non-government organizations as main source for data. So, data from them are assumed to be reliable and accurate. In addition, the policies and standards for using those APIs are assumed to be well-documented.

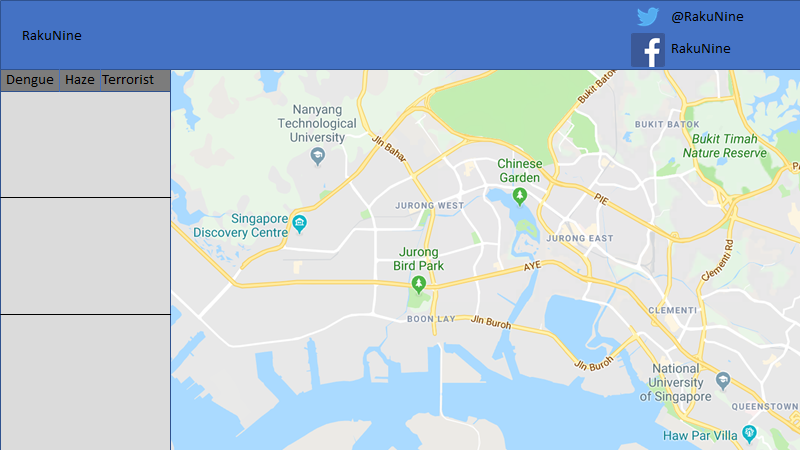
# External Interface Requirements

## User Interfaces

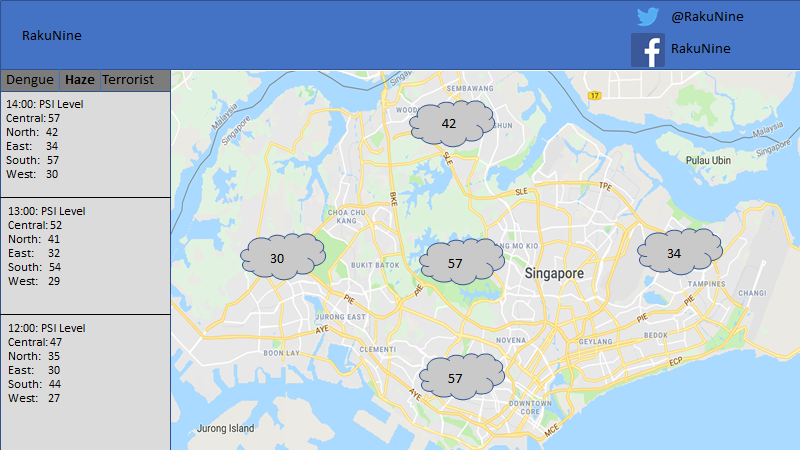
There are two interfaces we have designed to cater to the general public and the call center. Both interfaces are web applications that are required to support all types of web browsers.

General Public UI:

When the user goes into our web page, he should be able to see a map on the right and a crisis live feed on the right like shown below. Our social media links are also shown on this page so that users are able to receive any updates. On this page, the user will be able to see live feeds of the crisis information that we are providing. The user will also be able to filter what kind of crisis information they want to see. For example, if the user wants to check the haze conditions, they can click on the ‘Haze’ on the live feed.

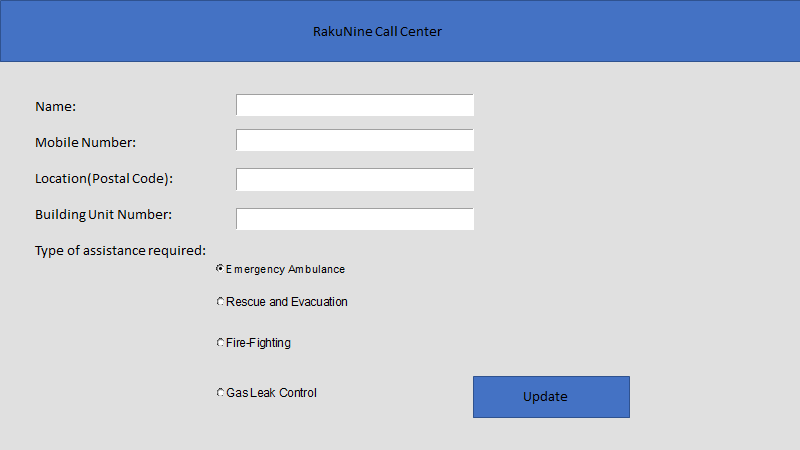


After the user clicks ‘Haze’ , there will be a hourly PSI level reading on the live feed as well as the PSI level reading on the map on five locations as shown below.



Call Center UI:

The call center operator will only be able to see a web page with an electronic form to update data that is being given to them by the general public.



## Software Interfaces

|  |  |
| --- | --- |
| Software Interface | Software Product Required |
| Operating System | Windows |
| Database System | MySQL |
| Front-End Implementation | HTML, CSS, Bootstrap, Php, Javascript |
| Back-End Implementation | C#, Java |

API Services Used

|  |  |
| --- | --- |
| Email | <https://developers.google.com/gmail/api/quickstart/dotnet> |
| Facebook | <https://developers.facebook.com/docs/pages/publishing> |
| Twitter | <https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/post-statuses-update> |
| SMS | <https://www.twilio.com/docs/sms/quickstart/csharp-dotnet-framework> |
| Dengue | <https://developers.onemap.sg/privateapi/themesvc/retrieveTheme?queryName=dengue_cluster&token=>  <https://www.nea.gov.sg/dengue-zika/dengue/dengue-clusters> |
| PSI | <https://data.gov.sg/dataset/psi?view_id=496c77eb-6add-4e9b-9883-17864cedfe9f&resource_id=82776919-0de1-4faf-bd9e-9c997f9a729d> |
| Weather | <https://data.gov.sg/dataset/realtime-weather-readings?view_id=b826c97c-d4f8-4e3e-9cc1-110a5281d30f&resource_id=17494bed-23e9-4b3b-ae89-232f87987163> |

## Communications Interfaces

Our system is based on a client-server model where our users will be able to use our system via HTTP/S. Our system should be accessible using modern web browsers. Some other communication modes our system use are E-mail and SMS.

# System Features

## Provide Real-time Crises Information on Map

* + 1. Description and Priority

The user checks on real time crisis information (High Priority)

* + 1. Stimulus/Response Sequences

1. The system displays the map interface
2. The system retrieves information from database
3. The system displays lists of current events based on tiers
   * 1. Functional Requirements
4. The user must be able to view a map of Singapore with real time crisis information
   1. The system must be able to display a map of Singapore
   2. The system must be able to retrieve real time crisis information from database
   3. The system must be able to displays real time crisis information based on tiers

## Generate Status Report for Prime Minister

* + 1. Description and Priority (Medium Priority)

The system generates a summary report and deliver it to the PM office every 30 minutes

* + 1. Stimulus/Response Sequences

1. The system retrieves the latest information from the database
2. The system summarizes the information based on tiers and trends
3. The system generates the report
4. The system uses an external API to send an email to the PM’s office
   * 1. Functional Requirements
5. The Prime Minister must be able to receive a status report summarizing key indicators and trends over email every 30 minutes
   1. The system must be able to retrieve the latest information from the database
   2. The system must be able to summarize the information based on tiers and trends
   3. The system must be able to generate the report
   4. The system must be able to use an external API to send an email to the PM’s office

## Call Center Update

* + 1. Description and Priority

Call Center operator updates system based on calls from General Public (High Priority)

* + 1. Stimulus/Response Sequences

1. The system displays the update info interface
2. The operator enters the name into the name textbox
3. The operator enters the into the mobile number textbox
4. The operator enters the location information into the location textbox
5. The operator selects the type of assistance requested
6. The operator enters the assistance information into the assistance textbox
7. The operator clicks the submit button
8. The system verifies the information
9. The system displays a confirmation message
10. The system users the included use case “Differentiate Tiers” to sort the information
11. The system updates the database
12. The system uses the included use case “receive dispatch command” to send out SMS to relevant agencies
    * 1. Functional Requirements
13. The operator must be able to enter the name of the caller in the name textbox
    1. The system must be able to display error messages when operator does not input anything in name textbox
14. The operator must be able to enter the mobile number of the caller in the mobile number textbox
    1. The system must be able to display error messages when operator does not input anything in mobile number textbox
15. The operator must be able to enter the location information in the location textbox
    1. The system must be able to display error messages when user does not input anything in location textbox
16. The operator must be able to select an assistance type from a list of all type of assistance
    1. The system must be able to display error messages when operator does not select an assistance type
17. The operator must be able to enter the assistance information in the assistance textbox
    1. The system must be able to display error messages when operator does not input anything in assistance textbox

## Update Real-Time Data with API

* + 1. Description and Priority

The system gets real time information from the different APIs (High Priority)

* + 1. Stimulus/Response Sequences

1. The system retrieves data from the APIs
2. The system updates the database
   * 1. Functional Requirements
3. The System must be able to update database with data from APIs
   1. The System must be able to retrieve data from APIs

## 

## Update Social Media

* + 1. Description and Priority

The system updates public on useful information through social medias periodically

(Medium Priority)

* + 1. Stimulus/Response Sequences

1. The system retrieves information from the database
2. The system summarizes the information
3. The system generates a social media post based on the information
4. The system pushes an update through the social media API
   * 1. Functional Requirements
5. The system must be able to generate a social media post based on information from the database
   1. The system must be able to retrieve information from database
   2. The system must be able to summarize the information retrieved from database
   3. The system must be able to generate a social media post with the summarized information
   4. The system must be able to push the social media post through the social media API

## Dispatch Command with SMS

* + 1. Description and Priority

The system sends an SMS to dispatch the relevant agencies (High Priority)

* + 1. Stimulus/Response Sequences

1. The system retrieves information from database
2. The system generates the message to be sent
3. The system retrieves the relevant agencies information
4. The system uses SMS API to send message to relevant agencies
   * 1. Functional Requirements
5. The system must be able to send an SMS to dispatch the relevant agencies with informations from the database
   1. The system must be able to retrieve information from database
   2. The system must be able to generate a message with information from the database
   3. The system must be able to retrieve information about the relevant agency from database
   4. The system must be able to use SMS API to send the generated message to the relevant agencies

# Other Nonfunctional Requirements

## Security

* Front-end web pages and back-end database server must be able to withstand server overrides.
* All api requests must be able to withstand data sniffing.
* The system must be able to withstand and response to SQL Injection attack.

## Flexibility

* New APIs must be able to be added to the system without modifying the main source code of the system.