Component Implementation Documentations and Test Cases

By

Team RakuNine

Table of Contents

[FrontEnd Components 3](#_Toc528327801)

[Public UI 3](#_Toc528327802)

[Description 3](#_Toc528327803)

[Test Cases 3](#_Toc528327804)

[Result 4](#_Toc528327805)

[Call Center UI 4](#_Toc528327806)

[Description 4](#_Toc528327807)

[Test Cases 4](#_Toc528327808)

[Result 5](#_Toc528327809)

[BackEnd Components 6](#_Toc528327810)

[Weather Sub-System 6](#_Toc528327811)

[Description 6](#_Toc528327812)

[Test Cases 6](#_Toc528327813)

[Result 8](#_Toc528327814)

[PSI Sub-System 8](#_Toc528327815)

[Description 8](#_Toc528327816)

[Test Cases 8](#_Toc528327817)

[Result 9](#_Toc528327818)

[SMS Sub-System 9](#_Toc528327819)

[Description 9](#_Toc528327820)

[Test Cases 9](#_Toc528327821)

[Emergency Sub-System 10](#_Toc528327822)

[Description 10](#_Toc528327823)

[Test Cases 10](#_Toc528327824)

[Result 14](#_Toc528327825)

[Dengue Sub-System 14](#_Toc528327826)

[Description 14](#_Toc528327827)

[Test Cases 14](#_Toc528327828)

[Result 14](#_Toc528327829)

[Social Media Sub-System 15](#_Toc528327830)

[Description 15](#_Toc528327831)

[Test Cases 15](#_Toc528327832)

[Email Sub-System 15](#_Toc528327833)

[Description 15](#_Toc528327834)

[Test Cases 15](#_Toc528327835)

[Main Controller 15](#_Toc528327836)

[Description 15](#_Toc528327837)

[Test Cases 16](#_Toc528327838)

# FrontEnd Components

There are two frontend main components, Public UI and Call Center UI. They are built by react and tested by black box testing method and unit testing method.

## Public UI

### Description

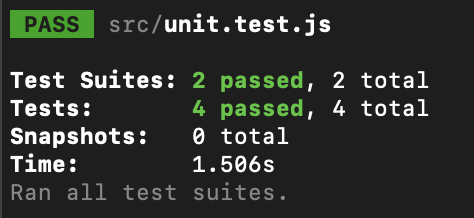
Public UI component is built by using react. So, it is very interactive. Values are stored in states. So, they can be refreshed without refreshing the browsers. Generally, it will asynchronously fetch the data from server and display it on the map and in the live feed.

### Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Description | Expected Output | Actual Output | Result |
| 1. | When the use clicks the dengue radio button | Data related to the dengue fever are displayed on the map and live feed | Data related to the dengue fever are displayed on the map and live feed | Pass |
| 2. | When the use clicks the haze radio button | Data related to the haze are displayed on the map and live feed | Data related to the haze are displayed on the map and live feed | Pass |
| 3. | When the use clicks the terrorist radio button | Data related to the terrorist attack are displayed on the map and live feed | Data related to the terrorist attack are displayed on the map and live feed | Pass |



### Result



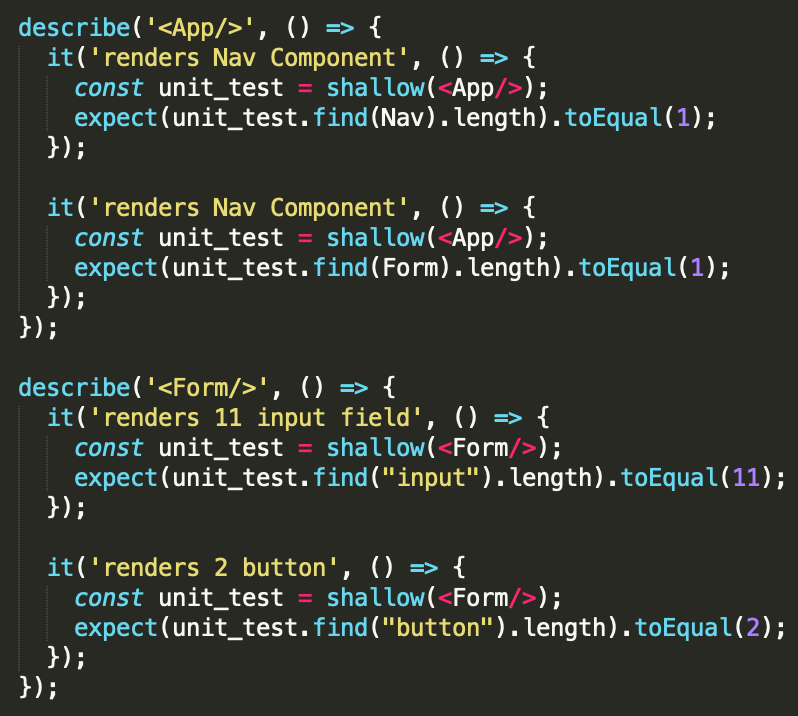
## Call Center UI

### Description

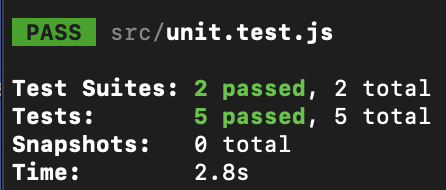
Call Center UI is built by using react. It is basically a form. When user key in the data into the form and click submit button, it will send data to the server.

### Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Description | Expected Output | Actual Output | Result |
| 1. | User puts in the Name | Data is stored in name state | Data is stored in name state | Pass |
| 2. | User puts in the Phone Number | Data is stored in phNumber state | Data is stored in phNumber state | Pass |
| 3. | User puts in the Location | Data is stored in location state | Data is stored in location state | Pass |
| 4. | User clicks emergency ambulance | Emergency Ambulance is stored in a data state | Emergency Ambulance is stored in a data state | Pass |
| 5. | User clicks rescue and evaluation | Rescue and Evaluation is stored in a data state | Rescue and Evaluation is stored in a data state | Pass |
| 6. | User clicks Fire Fighting | Fire Fighting is stored in a data state | Fire Fighting is stored in a data state | Pass |
| 7. | User clicks Gas Leak Control | Gas Leak Control is stored in a data state | Gas Leak Control is stored in a data state | Pass |
| 8. | User clicks the Submit button | Data are sent to the server | Data are sent to the server | Pass |



### Result



# BackEnd Components

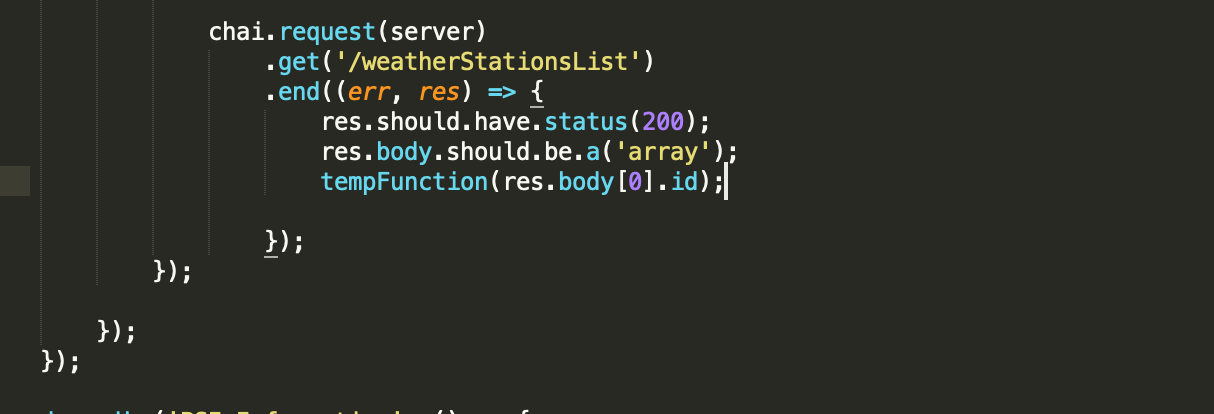
BackEnd Components are built by using node.js and they are tested by built-in code unit tests and black box testing method.

## Weather Sub-System

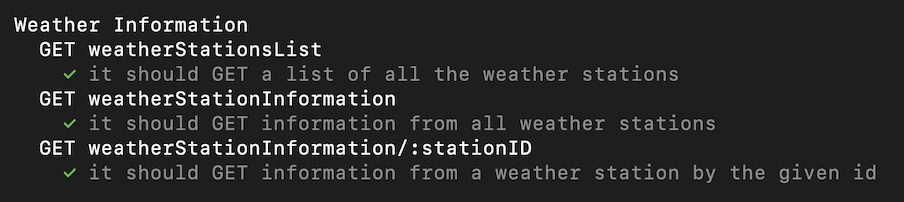
### Description

Weather component has Weather API Controller which communicates with Weather API. Weather Database Controller get the data from Weather API Controller and stored them inside the Metering Station Database and Weather Information Database. This component is used for handling weather related data.

### Test Cases



### Result



## PSI Sub-System

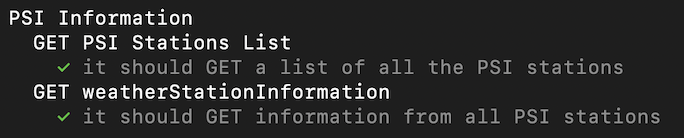
### Description

PSI component has PSI API Controller which communicates with PSI API. PSI Database Controller get the data from PSI API Controller and stored them inside the PSI Station Database and PSI Information Database. This component is used for handling PSI related data.

### Test Cases



### Result



## SMS Sub-System

### Description

SMS component has SMS controller which communicates with Main Controller and Emergency Controller. It also has SMS API controller which communicates with External SMS API. This component is used to send SMS to emergency services.

### Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Description | Expected Output | Actual Output | Result |
| 1. | The system receives the emergency request | It sends the SMS to emergency services | It sends the SMS to emergency services | Pass |
| 2. | The system receives the “case solved” reply from the emergency services | It updates the status of the case to “solved” | It updates the status of the case to “solved” | Pass |

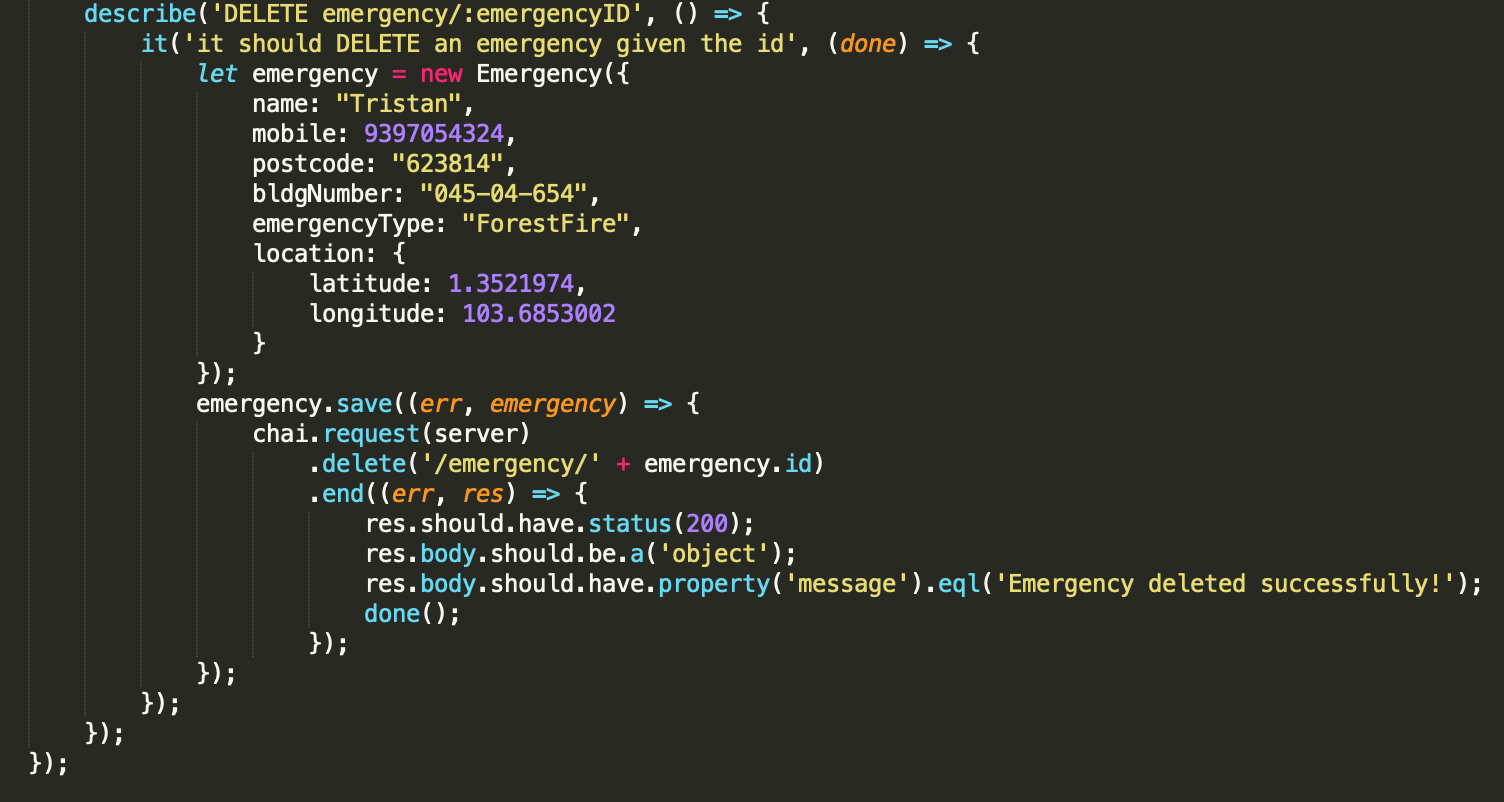
## Emergency Sub-System

### Description

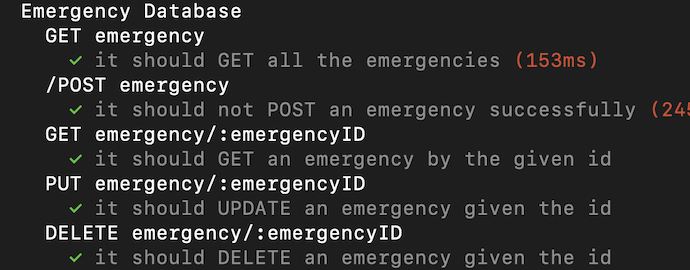
Emergency component has Emergency Controller which communicates with Main Controller and SMS Controller. Emergency Controller will get the data from Main Controller and stored them inside Emergency Database. It is used to handle all emergency related tasks.

### Test Cases





### Result

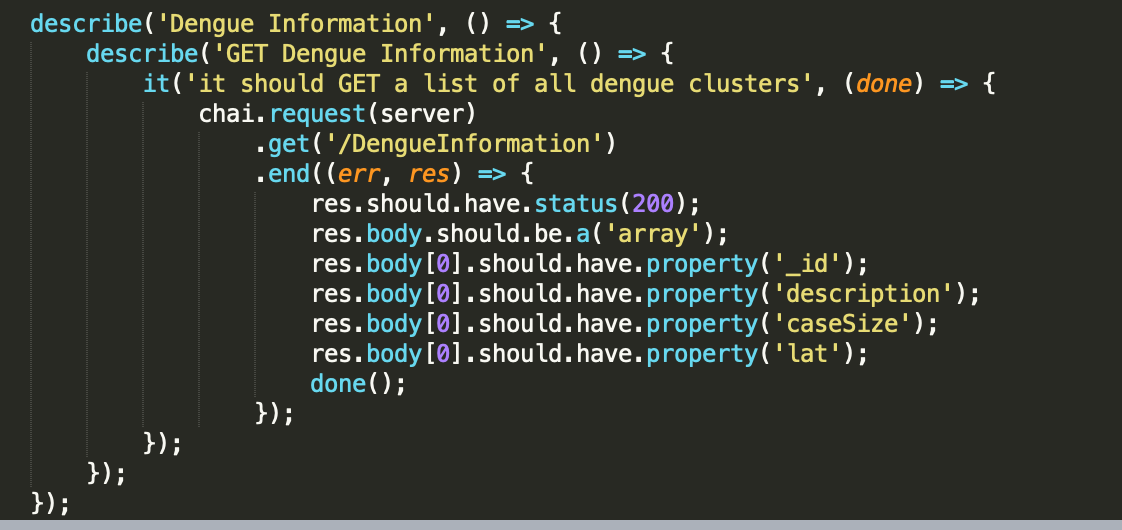


## Dengue Sub-System

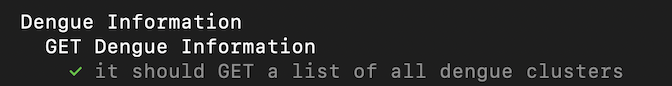
### Description

Dengue component has Dengue API Controller which communicates with Dengue API. Dengue Database Controller get the data from Dengue API Controller and stored them inside the Dengue Database. This component is used for handling dengue related data.

### Test Cases



### Result



## Social Media Sub-System

### Description

Social Media component has Social Media API controller which gets the data from Main Controller and send them to external Social Media API. This is used to post emergency status on Facebook and tweets on twitter.

### Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Description | Expected Output | Actual Output | Result |
| 1 | System want to post status on Facebook | Posted a status on a Facebook page | Posted a status on a Facebook page | Pass |
| 2 | System want to tweet on Twitter | Posted a tweet on Twitter | Posted a tweet on Twitter | Pass |

## Email Sub-System

### Description

Email component has Email API controller which will get the data from the Main Controller, create a report and send it to the government office via Email API. This is used to send status report to Government office every 30 minutes.

### Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Description | Expected Output | Actual Output | Result |
| 1 | System triggers generate report action | A report is generated | A report is generated | Pass |
| 2 | System sends email to government office | Email is sent and received at government office | Email is sent and received at government office | Pass |

## Main Controller

### Description

This component handles all the communications between backend components. This is the head component of the backend.

### Test Cases

This components do not need specific test cases. If all the above components are working fine, this main component is working fine.