Software Design Document
for
CCC’s Admin Panel for Policy Management System in AWS

Version 1.0
Prepared by Misael Corvera, Alexander Horejsi, Yi Wang, Jateni Dida, Zac You

Sponsored by Commonwealth Casualty Company
November 28th, 2019
# Table of Contents

Table of Contents............................................................................................................<pg 2>
Revision History..............................................................................................................<pg 4>
1. Introduction................................................................................................................<pg 5>
   1.1. Purpose............................................................................................................<pg 5>
   1.2. Document Conventions.....................................................................................<pg 5>
   1.3. Intended Audience and Reading Suggestions..................................................<pg 5>
   1.4. System Overview.............................................................................................<pg 5>
2. Design Considerations..................................................................................................<pg 6>
   2.1. Assumptions and dependencies.........................................................................<pg 6>
   2.2. General Constraints.........................................................................................<pg 6>
   2.3. Goals and Guidelines.......................................................................................<pg 6>
   2.4. Development Methods....................................................................................<pg 6>
3. Architectural Strategies..............................................................................................<pg 7>
4. System Architecture..................................................................................................<pg 8>
   4.1 DFD 0................................................................................................................<pg 8>
   4.2 DFD 1 ...............................................................................................................<pg 9>
5. Policies and Tactics....................................................................................................<pg 10>
   5.1. Specific Products Used.....................................................................................<pg 10>
   5.2. Requirements traceability................................................................................<pg 10>
   5.3. Testing the software.......................................................................................<pg 10>
6 Detailed System Design............................................................................................<pg 11>
   6.x Name of Module...............................................................................................<pg 11>
   6.x.1 Responsibilities.............................................................................................<pg 11>
   6.x.2 Constraints....................................................................................................<pg 11>
   6.x.3 Composition...................................................................................................<pg 11>
   6.x.4 Uses/Interactions.........................................................................................<pg 11>
   6.x.5 Resources.....................................................................................................<pg 11>
   6.x.6 Interface/Exports.........................................................................................<pg 11>
7 Detailed Lower level Component Design....................................................................<pg 14>
   7.x Name of Class or File.......................................................................................<pg 14>
   7.x.1 Classification..................................................................................................<pg 14>
   7.x.2 Processing Narrative(PSPEC).......................................................................<pg 14>
7.x.3 Interface Description............................................................................................................. <pg 14>
7.x.4 Processing Detail..................................................................................................................... <pg 14>
7.x.4.1 Design Class Hierarchy ...................................................................................................... <pg 14>
7.x.4.2 Restrictions/Limitations .................................................................................................... <pg 14>
7.x.4.3 Performance Issues ........................................................................................................... <pg 14>
7.x.4.4 Design Constraints ........................................................................................................... <pg 14>
7.x.4.5 Processing Detail for Each Operation ................................................................................ <pg 14>
8 Database Design ............................................................................................................................ <pg 18>
9 User Interface ............................................................................................................................... <pg 19>
10 Requirements Validation and Verification .................................................................................... <pg 26>
11 Glossary ........................................................................................................................................ <pg 28>
12 References ..................................................................................................................................... <pg 29>
## Revision History

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason for Changes</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Revision</td>
<td>11/28/19</td>
<td>Initial Document Write Up</td>
<td>1.0</td>
</tr>
</tbody>
</table>


1. Introduction

Commonwealth Casualty Company (CCC) was founded in 2010, and its mission is to provide affordable insurance to everyone in the community. Nearly 100 percent of the company’s dedicated employees volunteer their time with community service organization. CCC partners with several charitable organizations that help the community in need, and these include Habitat for Humanity, the Special Olympics, and so on. The company offers a full suit of affordable products and services which includes homeowners, renters, auto insurance, and roadside assistance. Most importantly, the company provides a variety of discounts to help people save more on insurance.

1.1 Purpose
The purpose of this document is to inform future developers about the tools used to create this software.

1.2 Document Conventions
Each requirement statement has its own priority.

1.3 Intended Audience and Reading Suggestions
This document is mainly for future developers that will work on this project to inform them about what went into making this software.

1.4 System Overview
The system shall save dashboard layout for each individual user when they login. Depending on the error level, the user shall be notified by the system with the error logs. A user shall have the ability to add/delete/modify graph panels, log display panels, as well as configure notification settings.
2. Design Considerations

2.1 Assumptions and Dependencies

The users of the CCC dashboard should be able to navigate through the system as the system’s functionalities are fairly clear.

2.2 General Constraints

The website design is used to lower hardware limitations. Anyone with IOS, Windows, Android, Linux, or any operating system that has a browser will be able to gain access to the website.

- Programming languages, frameworks, and paradigms are limited to: React, Springboot, MySQL
- Database Servers are required in the process. As the data increase performance speed also increases.
- Cost constraints are encountered as the project requires AWS resources
- Time constraints are encountered as the project must be completed in time
- Real time update of the metric data

2.3 Goals and Guidelines

Goals designs for this system software:

- An administrative dashboard that provides a snapshot report of overall system health
- A friendly user interface that provides drill-down features that are easy to navigate

2.4 Development Methods

Requirements: First step of our development is to understand what needs to be designed as well as the overall functionality requirements for the objectives of this project.

Design: System design is the second phase of our development. It is to design the system architecture that help us understand the flow of each functionality that we will be implementing.

Implementation: In the implementation stage, we divide the project into smaller modules and distribute each module to individual member accordingly to make progress.

Testing: After implementation stage, we will get to testing phase where we combine all the modules and test them to make sure that there won’t be any errors or flaws.

Deployment of System: After the testing phase, the software, or the product will be integrated into the customer environment and will be ready to use.

Maintenance: The maintenance phase will occur after delivery of the final product, which it will require regular use of the system and make changes to the system once the error is detected.
3. Architectural Strategies

- Java Spring boot will be used for the backend of this software since that is what the CCC system can support.
- MySQL will be used to retrieve all the stored information from the RDS server.
- React.js will be used for building the frontend UI (User Interface)

4. System Architecture

Frontend: React.js, React-boot-strap, Chart.js
Backend: Java Spring boot

4.1 Level 0 DFD

This DFD describes a rough view of how the system will handle requests from the user and its response.
4.2 Level 1 DFD

```
SNS

Lambda Function Triggers AWS notification service

SNS push the Message to User

Search

Make Inputs into Search Query

Retrieve Logs

Send Query

User Specified Metric Data

User Configuration

Required Selected Data Fields

Display Log Tables

Required Logs

Display Metric Data

User Name and Password

Login

Edit Layout

New Layout Information

User Specified Dashboard Fields

Draw Dashboard

Changing Layout

Populate Metric Data in Dashboard

Display Tables

Populate log tables in Dashboard

Retrieved Data

Specified User Data Fields

Requirement Check Flag
If meets, run Lambda Function

CloudWatch
```
5. Policies and Tactics

5.1 Choice of which specific products used
Multiple code editors and IDEs (integrated development environments) such as Microsoft Visual Studio, Sublime and Atom text editors were used in the development process. Javascript’s framework, React was used as a primary front end development language with multiple libraries integrated in the development process. Chart.js and ReactTable were the main two libraries we used to display the contents in the front end. Elegance and simplicity were the reason why we choose those two libraries and also they were easy to integrate with the way we grab data from the api calls.

AWS CLoudwatch insight to search and filter query results, AWS EC2 instances to run a live server on cloud. AWS Lambda to trigger AWS SNS (Simple Notification System) service whenever a change a user wants to know happens.

5.2 Plans for ensuring requirements traceability
Section 10 has a table which describes all of the requirements and which modules satisfy them

5.3 Plans for testing the software
We plan to have the CCC technical stuff use and test the new system health management system and get feedback. Based on their feedback, we can modify the monitoring system to ensure a smooth user-experience.

6. Detailed System Design

<table>
<thead>
<tr>
<th>6.1 Add New Graph/Table</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1.1 Responsibilities</td>
<td>This module is responsible for adding new graph types in the dashboard</td>
</tr>
<tr>
<td>6.1.2 Constraints</td>
<td></td>
</tr>
<tr>
<td>6.1.3 Composition</td>
<td></td>
</tr>
<tr>
<td>6.1.4 Uses/Interactions</td>
<td>This module is meant to interact with the Display module.</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>6.1.4 Resources</td>
<td></td>
</tr>
<tr>
<td>6.1.5 Interface/Exports</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6.2 Display real time Table</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.1 Responsibilities</td>
<td>This module is responsible for displaying real time percentage of log errors in a log group of a particular instance</td>
</tr>
<tr>
<td>6.2.2 Constraints</td>
<td></td>
</tr>
<tr>
<td>6.2.3 Composition</td>
<td></td>
</tr>
<tr>
<td>6.2.4 Uses/Interactions</td>
<td>Almost all other modules will depend on this module to perform CRUD operations.</td>
</tr>
<tr>
<td>6.2.4 Resources</td>
<td></td>
</tr>
<tr>
<td>6.2.5 Interface/Exports</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6.3 Edit Graph/Table</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3.1 Responsibilities</td>
<td>This module is responsible for editing the newly added graph or table.</td>
</tr>
<tr>
<td>6.3.2 Constraints</td>
<td></td>
</tr>
<tr>
<td>6.3.3 Composition</td>
<td></td>
</tr>
<tr>
<td>6.3.4 Uses/Interactions</td>
<td>This module is meant to interact with the Display module.</td>
</tr>
<tr>
<td>6.3.4 Resources</td>
<td></td>
</tr>
<tr>
<td>6.3.5 Interface/Exports</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>6.4 Delete Graph/Table</td>
<td></td>
</tr>
<tr>
<td>6.4.1 Responsibilities</td>
<td>This module is responsible for deleting unwanted graph or table in the dashboard</td>
</tr>
<tr>
<td>6.4.2 Constraints</td>
<td></td>
</tr>
<tr>
<td>6.4.3 Composition</td>
<td></td>
</tr>
<tr>
<td>6.4.4 Uses/Interactions</td>
<td>This module is meant to interact with the Display module.</td>
</tr>
<tr>
<td>6.4.4 Resources</td>
<td></td>
</tr>
<tr>
<td>6.4.5 Interface/Exports</td>
<td></td>
</tr>
<tr>
<td>6.5 Subscribe to notification</td>
<td></td>
</tr>
<tr>
<td>6.5.1 Responsibilities</td>
<td>This module is responsible for subscribing users to a specific topic they choose or be alerted of a specific event.</td>
</tr>
<tr>
<td>6.5.2 Constraints</td>
<td></td>
</tr>
<tr>
<td>6.5.3 Composition</td>
<td></td>
</tr>
<tr>
<td>6.5.4 Uses/Interactions</td>
<td>This module is meant to interact with the Display module.</td>
</tr>
<tr>
<td>6.5.4 Resources</td>
<td></td>
</tr>
<tr>
<td>6.5.5 Interface/Exports</td>
<td></td>
</tr>
</tbody>
</table>
7. Detailed Lower level Component Design

7.1 Update
   7.1.1 Class
   7.1.2 Used to change a user’s dashboard via adding, removing and modifying

7.2 Add
   7.2.1 Class
   7.2.2 Used to add graphs to a user’s dashboard

7.3 Remove
   7.3.1 Class
   7.3.2 Used to delete a graph from the dashboard

7.4 Modify
   7.4.1 Class
   7.4.2 Used to change the content of one of a user’s graphs
   7.4.3 No interface.

7.5 UpdateController.java
   7.5.1 Class
   7.5.2 Performs updating of dashboard
   7.5.3 No interface.

7.6 AddController.java
   7.6.1 Class
   7.6.2 Adds graph to dashboard
   7.6.3 No interface

7.7 RemoveController.java
   7.7.1 Class
   7.7.2 Removes graph from user’s dashboard
   7.7.3 No interface

7.8 ModifyController.java
   7.8.1 Class
   7.8.2 Modifies graph in dashboard
7.8.3 No interface

7.9 Bootstrap, React and Chart.js

7.9.1 Package

7.9.2 Provides all the necessary components that make up the application’s UI.

8. Database Design

9. User Interface

9.1 Overview of User Interface

When the application is booted up, the first thing the user will see is the login page. After successfully logging in the user will be redirected to the dashboard page. In the dashboard page, there will be tables and charts in addition to the menu bar that can be found at the left side of the page and the search bar located at the top of the page. In the menu bar, there will be links that will call the form component in order to add new charts or tables to the dashboard.

9.2 Screen Frameworks or Images
9.2.1 Login Page

9.2.2 Home Page
9.2.3 Form Component

9.3 User Interface Flow Model
10. Requirements Validation and Verification

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Component/Module</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.1 Application Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.1.1.1 Check user input for login:</td>
<td>Login.js</td>
<td>Observation of Code and Manual Testing</td>
</tr>
<tr>
<td>10.1.1.2 Check database for role credentials:</td>
<td>Authenticate.java and MySQL</td>
<td>Observation of Code and Manual Testing</td>
</tr>
<tr>
<td>10.1.1.3 Redirect to home page and create user’s dashboard.</td>
<td>AutLayout.js and dashboard.js</td>
<td>Observation of Code and Manual Testing</td>
</tr>
<tr>
<td>10.1.4 From dashboard page user has access to navigation bar and dashboard.</td>
<td>AutLayout.js and dashboard.js</td>
<td>Observation of Code and Manual Testing</td>
</tr>
<tr>
<td>10.1.5 From navigation bar user can do the following actions:: Add real time chart. Add Log Table. Add charts. Modify Alerts. Access Settings.</td>
<td>AutLayout.js and sidebar.js (uses BarGraph.js, LineGraph.js, Table.js, and RightSidebar.js)</td>
<td>Observation of Code and Manual Testing</td>
</tr>
</tbody>
</table>
10.1.6 From the top bar the user can do the following:
Make a search.
Check notifications.

<table>
<thead>
<tr>
<th>10.1.6 From the top bar the user can do the following:</th>
<th>AutLayout.js and Topbar.js (uses NotificationDropdown.js and Search.js)</th>
<th>Observation of Code and Manual Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.2 Logout</td>
<td>Logout.js</td>
<td>Observation of Code and Manual Testing</td>
</tr>
</tbody>
</table>
11. Glossary

**AWS**: Amazon Web Services.

**AWS EC2**: Elastic Compute Cloud

**AWS SNS**: Simple Notification Service

**AWS Lambda**: Serverless Computer

**CCC**: Commonwealth Casualty Company

**MySQL**: an open source relational database management system.

**RESTful API**: an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.

**RDS**: Amazon Relational Database Service

**VPC**: Virtual private cloud