Software Requirements Specification

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............................................................................................................................................... <pg 3>
1. Introduction............................................................................................................................ <pg 4>
   1.1. Purpose......................................................................................................................... <pg 4>
   1.2. Intended Audience and Reading Suggestions...................................................... <pg 4>
   1.3. Product Scope.......................................................................................................... <pg 4>
   1.4. Definitions, Acronyms, and Abbreviations......................................................... <pg 4>
   1.5. References............................................................................................................... <pg 4>
2. Overall Description.................................................................................................................. <pg 6>
   2.1. Product Perspective.................................................................................................. <pg 6>
   2.2. Product Functions..................................................................................................... <pg 6>
   2.3. User Classes and Characteristics............................................................................... <pg 6>
   2.4. Operating Environment............................................................................................ <pg 6>
   2.5. Design and Implementation Constraints.............................................................. <pg 7>
   2.6. User Documentation..................................................................................................... <pg 7>
   2.7. Assumptions and Dependencies................................................................................. <pg 7>
   2.8. Apportioning of Requirements................................................................................. <pg 7>
3. External Interface Requirements............................................................................................. <pg 8>
   3.1. User Interfaces............................................................................................................ <pg 8>
   3.2. Hardware Interfaces................................................................................................... <pg 8>
   3.3. Software Interfaces..................................................................................................... <pg 8>
   3.4. Communications Interfaces....................................................................................... <pg 8>
4. Requirements Specification........................................................................................................<pg 9>
   4.1. Functional Requirements............................................................................................. <pg 9>
   4.2. External Interface Requirements............................................................................... <pg 9>
   4.3. Logical Database Requirements................................................................................ <pg 10>
   4.4. Design Constraints..................................................................................................... <pg 10>
5. Other Nonfunctional Requirements.........................................................................................<pg 11>
   5.1. Performance Requirements....................................................................................... <pg 11>
   5.2. Safety Requirements.................................................................................................. <pg 11>
   5.3. Security Requirements............................................................................................... <pg 11>
   5.4. Software Quality Attributes...................................................................................... <pg 11>
   5.5. Business Rules........................................................................................................... <pg 12>
6. Other Requirements.................................................................................................................<pg 13>
Revision History

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason For Changes</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft # 1</td>
<td>11/28/2019</td>
<td>First draft</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason For Changes</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Introduction
This document is a Software Requirement Specification (SRS) for the Commonwealth Casualty Company (CCC)'s Admin Panel for Policy Management System in AWS. The goal of this application is to provide an administrative dashboard for CCC that provides a snapshot report of overall system health.

1.1 Purpose
The purpose of this document is to inform readers about CCC’s system health management system dashboard and its uses. It will illustrate the purpose and declaration of definitions, abbreviations, every version numbers before and after revisions. This document is primarily intended for CCC developers and employees in general, this will help create the first version, which is a reference to the web application for the development team. The system is meant to create a visual representation and intuitive way to display the overall health status of all virtual private clouds and servers running on Amazon Web Services and other cloud platforms.

1.2 Intended Audience and Reading Suggestions
This document is meant to be read by the software developers of CCC. For a general idea of how the software works, please check Section 3.

1.3 Product Scope
The new CCC’s system health management dashboard will provide an extensive overview of the overall system and all Virtual private clouds’s health status in an intuitive and more user friendly way for employees of CCC. Authorized employees of CCC can monitor the overall system health in real time and also they will be able to navigate to the virtual system for a fix in error or any desired activity provided by CCC system health dashboard.

Users also can monitor multiple metrics of the services provided by Amazon Web Services (AWS). These intended users also can import and populate the default dashboard with graphs and tables for multiple monitoring of the system.

As an authorized user, the intended user should have an access ID and secret access id to be able to access these resources provided by CCC system health dashboard.

1.4 Definitions, Acronyms, and Abbreviations
All definitions, acronyms, and abbreviations are listed at the end of the document.

1.5 References

All references are listed at the end of the document.
2. Overall Description

This section gives background information about specific requirements of the web application service to be developed in brief. Although we will not describe every requirement in detail, this section will describe the factors that affect the final product.

2.1 Product Perspective

This software product is intended for CCC employees. The product will be deployed on the AWS for all employees to have access to. There will be a database on cloud server provided by AWS where all the data is kept and managed.

To use the application users must be assigned a username and password by CCC. Users will be able to login and logout of the system anytime he/she wants.

This software is running in corporate with some other application run by the company for monitoring purpose. This software aims to be a data aggregator on top of the existing infrastructure with an easy to use and customizable UI.

2.2 Product Functions

This software aims to provide user easy-accessible data about the server and application CCC is using. It will provide the metric data about the server both in data points and graph in real time. It will also provide the user all the logs being collected and search for desired logs for troubleshooting and maintaining purpose. It will also notify users via e-mail according to their own setting. And it is designed to have a customized dashboard so the users would see only the data they want to see.

2.3 User Classes and Characteristics

Users of this web application will be CCC IT staff. It is safe to assume they would need easy access to the data this application provided for quick response and easier diagnostic of their system.

2.4 Operating Environment

This software will be running on AWS cloud server and is accessible through most of the browser either on phone or computer.
2.5 Design and Implementation Constraints
This software is built using many AWS functionality to cut down production time and difficulty and optimized the usage of AWS to cut down the operational expanse.

2.6 User Documentation
This software will not include any user documentation.

2.7 Assumptions and Dependencies
One assumption about this application is that all the logs being collected is pre-formatted for the search function to work correctly.

2.8 Apportioning of Requirements (pending)
TBD
3. External Interface Requirements

3.1 User Interfaces
The current system has one web application with two different view configurations, one on which the user has full access to all the functionalities of the dashboard, and another one where the dashboard will display the current charts and tables on the dashboard only. The main view has the ability to add or delete new charts or tables, depending on the type of chart, there is a form that needs to be filled out in order to get data from AWS CloudWatch. It also has a login page in order to authenticate the user. When the user logs in, he or she will be able to perform actions such as search logs, add charts and tables, and set up email notifications. As of right now, the main page will display a sample user’s view of the system. It shall have buttons and links that direct the user to the set pages and functions.

3.2 Hardware Interfaces
A user must use either a mobile device, a tablet, a laptop, or a desktop connected to the private cloud network of the company to get access to the web application. Using any internet browser such as Chrome or Firefox, the user shall be able to enter the website address which is managed by a web server and access the application.

3.3 Software Interfaces
The front end software will be written in JavaScript. The software will be using Node.js for package management and React.js to create components and pages. The back end software will be written in java. The software will be using MySQL, Maven and Spring Boot to create a RESTful API to retrieve and store the user’s data. The lambda function that the system is using to send email notifications will be also written in java.

3.4 Communications Interface
The dashboard will communicate with the RESTful API which will then communicate with the database in order to retrieve or save the user’s configurations. The dashboard will also communicate with AWS CloudWatch by using AWS SDK using asynchronous functions. There will also be timers on certain functions in order to display real time metrics on the dashboard.
4. Requirements Specification

4.1 Functional Requirements

The system shall:
4.1.1) Allows and checks user inputs:
4.1.1.1) Checks for login credentials and allows log out
4.1.1.2) Allows user to add/remove/modify graph panels based on inputs
4.1.1.3) Allows user to add/remove log tables based on inputs
4.1.2) Saves data on AWS RDS server
4.1.3) Pass in parameter
4.1.3.1) Using functions provided by AWS SDK
4.1.3.2) Get Parameter from user inputs
4.1.4) Allows query search for logs
4.1.5) Allows notification configuration
4.1.5.1) Allows user to turn on/off notification triggered by levels
4.1.5.2) Allows user to be notified by emails/phones

4.2 External Interface Requirements

4.2.1) This system shall allow user input as type of input to the system.

It contains both content and format as follows:
- Name of item
- Description of purpose
- Source of input or destination of output
- Valid range, accuracy and/or tolerance
- Units of measure
- Timing
- Relationships to other inputs/outputs
- Screen formats/organization
- Window formats/organization
- Data formats
- Command formats
- End messages
4.3 Logical Database Requirements

This may include:

- A user table which stores:
  - Id
  - Name
  - Email
  - Dashboard json
  - Notification setting
- Integrity constraints
- Data retention requirements

(Please be advised) The security and assets schema will all be located in a different database than the main system. Also, all web service calls will be requested to and from there as well.

See appendix B for images of the assets table ERD and the security ERD.

4.4 Design Constraints

- The dashboard is website design 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, MySql, Springboot and AWS.
- Database server is required.
- Time constraints, project must be completed in time.
- Cost constraints, project requires AWS resources which will be provided by liaisons
- Real time update of the data
5. Other Nonfunctional Requirements

5.1 Performance Requirements

- For web-based application, the general rule is: 0-0.5 seconds is good, 0.5-2 seconds is acceptable
- All AWS Lambda functions shall use Node.js or Java, whichever is faster

5.2 Safety Requirements

There shall be a login feature for this app. Each login shall exist for an employee of Commonwealth Casualty Company. There shall not be a sign up feature.

5.3 Security Requirements

This product shall have a login verification for users. The application software safety requirements shall check for user rights when logging into the application. There shall not be an option for creating a new account because this application exists to only be used internally by Commonwealth Casualty Company.

5.4 Software Quality Attributes

This product shall be available on laptop and desktop. This application shall send emails whenever a log that meets certain criteria is created.

5.5 Business Rules

All users of this application have the same rights. The only difference between users is the layout of their dashboard which will be customizable.
Appendix A: Glossary

**AWS**: Amazon Web Services.

**AWS EC2**: Elastic Compute Cloud

**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
Appendix B: Analysis Models

Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.

Data Process Flow
Database Relational Diagram

- Username
- Email
- ID
- Dashboard JSON

Users
Appendix C: To Be Determined List