Data Analytics

**(Project Title)**

Software Design Document

Name: Mike Maryn

Date: 08/08/2016

**TABLE OF CONTENTS**

**1. INTRODUCTION 3**

1.1 Purpose 3

1.2 Scope 3

1.3 Overview 3

1.4 Reference Material 3

1.5 Definitions and Acronyms 3

**2. SYSTEM OVERVIEW 4**

**3. SYSTEM ARCHITECTURE 5**

3.1 Architectural Design 5

3.2 Decomposition Description 5

3.3 Design Rationale 5

**4. DATA DESIGN 6**

4.1 Data Description 6

4.2 Data Dictionary 6

**5. COMPONENT DESIGN 7**

**6. HUMAN INTERFACE DESIGN 8**

6.1 Overview of User Interface 8

6.2 Screen Images 8

6.3 Screen Objects and Actions 8

**7. REQUIREMENTS MATRIX 9**

**8. APPENDICES 9**

**1. INTRODUCTION**

**1.1 Purpose**

This design document describes the architecture and system design of XX. …. XX purpose is to allow for real-time search through the large amounts of data gathered by the MedVoice app, and display it in a readable manner for business purposes.

**1.2 Scope**

To easily display log data, which can be difficult to decipher due to a lack of standardization between databases (SQL, NoSQL, ElasticSearch, mongo DB, etc.) and make it easy to understand for business/analytic purposes. The software will take in mock user data (such as name, address, health rating, IP, alignment), store them in a generalized format and allow that data to be queried.

**1.3 Overview**

This document will contain the purpose, scope, system overview, system architecture, data design, and component design of this project.

**1.4 Reference Material**

List any documents, if any, which were used as sources of information for the test plan.

<https://www.sitepoint.com/building-recipe-search-site-angular-elasticsearch/>

<https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html>

<https://www.elastic.co/guide/en/logstash/current/introduction.html>

<https://www.elastic.co/guide/en/kibana/current/getting-started.html>

**1.5 Definitions and Acronyms**

Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SDD. These definitions should be items used in the design document that are most likely not known to the audience.

|  |  |
| --- | --- |
| SRS | Software Requirements Specification |

**2. SYSTEM OVERVIEW**

The project will allow a user to query through a database, using a simple, easy to follow method that will require no prior knowledge of the database to use. This will be down by storing the data into ElasticSearch and user AngularJS to query through it. The user will input his or her search term through a webpage designed with HTML and CSS.

**3. SYSTEM ARCHITECTURE**

**3.1 Architectural Design**

C:\Users\Mike\Downloads\Untitled Diagram (5).png

Data will be piped from \*\*\*source\*\*\* to ElasticSearch. Kibana will connect to ElasticSearch and visual some of that data. The website server will take queries from the user via the website and grab information; both text and visual form, then display it.

**3.2 Decomposition Description**

C:\Users\Mike\Downloads\Untitled Diagram (4).pngBasic Website Layout

**3.3 Design Rationale**

Discuss the rationales for selecting the architecture described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn’t choose them.

HTML provides an easy user interface, AngularJS easy to query from ElasticSearch, and ElasticSearch will not as easy to learn as some other database was built with the idea of searching.

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributes** | **weight** | **MySQL** | **ElasticSearch** |
| scalability | X.3 | 2 | 8 |
| Open Source | x.2 | 10 | 10 |
| Search | x.2 | 5 | 9 |
| Flexibility | X.1 | 4 | 7 |
| Server OS | X.1 | 7 | 9 |
| Easy of Code | x.1 | 10 | 5 |
| **Total** | **X1.0** | **5.7** | **8.3** |

**4. DATA DESIGN**

**4.1 Data Description**

Explain how the information domain of your system is transformed into data structures.

Describe how the major data or system entities are stored, processed and organized. List any databases or data storage items.

**4.2 Data Dictionary**

Alphabetically list the system entities or major data along with their types and descriptions.

AngularJS – JavaScript Framework

CSS – Style Sheet

ElasticSearch – Search Engine and Storage

Html – Markup Language

Kibana – Visualization of Data

Logstash – Data Pipeline

**5. COMPONENT DESIGN**

In this section, we take a closer look at what each component does in a more systematic way.

AngularJS – Connects the Website to ElasticSearch, allows for the programmer to write in the basic structure of the search query so the user doesn’t need to. (User fills in the blanks)

CSS – Makes the Website look pretty

ElasticSearch – Stores the data and query of data. Due to the popularity of the ELK stack it is easy to integrate with Logstash and Kibana

HTML – Creates the skeleton of the website

Kibana – Allows for pie chart, graphs, etc. of data to be created, easily integrated with ElasticSearch.

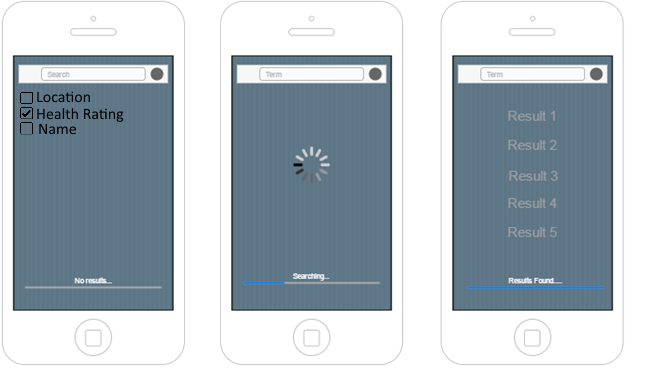
Logstash – Moves data from one source to another, easily integrated with ElasticSearch.

**6. HUMAN INTERFACE DESIGN**

**6.1 Overview of User Interface**

User will be present a screen featuring a search bar and several checkboxes. The search bar will be where the user will input a term they wish to search, such as a name, alignment, location. The checkboxes will allow the user to limit where the app searches for that term, again such as name, alignment, location. Then the user will press a button to start the search and the app will return a list of stored data, along with relevant graphs, and data such as number of results returned.

**6.2 Screen Images**



**6.3 Screen Objects and Actions**

Screen has three objects, a textbox, a series of checkboxes and a search button. The user will input a term to search, check boxes to limit where the app should look for the term, and search button to initiate the search.

**7. REQUIREMENTS MATRIX**

Provide a cross reference that traces components and data structures to the requirements in your requirements document.

Use a tabular format to show which system components satisfy each of the functional requirements from the requirements document. Refer to the functional requirements by the numbers/codes that you gave them in the requirements.

ElasticSearch = Database

Apache = Webserver

AngularJS = Backend

HTML = Frontend

**8. APPENDICES**

*This section is optional.*