**Software Requirements Specification**

**for**

**Mapping for Change Graph Generator**

**Rev. 1.3**

**Prepared by Reece Walsh**

**UBC Okanagan**

**Feb 5th 2018**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Features 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Reece Walsh | 21/2/18 | Initial version | 1.0 |
| Jeff Thomson | 21/2/18 | Added revision history | 1.1 |
| Daniel Herman | 23/3/18 | Edited our requirements to better reflect our clients wishes. Added some suggestions | 1.2 |
| Zachary Bentsen | 1/4/18 | Clarified versioning and rephrased selected sections. | 1.3 |

1. **Introduction**
   1. **Purpose**

The Mapping for Change Report Generator is designed to facilitate the online or offline viewing and creation of insightful reports for the end user. This document refers to the pre-release variant of the project. This document’s scope will include requirements for a website that will run in the end users browser, a PHP REST API for data transmission, and a database system that will serve graph data.

* 1. **Intended Audience and Reading Suggestions**

This Software Requirements document is intended for developers, project managers, website administrators, and devOps. This document is best read in chronological order or as a referential text.

* 1. **Product Scope**

The purpose of this software is to provide community planners with complex analysis tools for data on community service establishments. The goal is to give information that matters to the people who need it and use it so that they may help as many people as possible.

* 1. **References**

The data we are using is a snapshot from the <https://isearchkelowna.ca/> database, courtesy of our client.

1. **Overall Description**
   1. **Product Perspective**

This product is designed as a module for the Mapping for Change Initiative. It allows the users to analyze the data that the site provides, so that city planners can see where resources are needed. The goal of this product is to get more people on board with integrating data collection from services that help the vulnerable by showing them what is possible.

This product should easily integrate with the existing mapping for change website. There is no direct interaction between the two systems, however, they both rely on the same database.

* 1. **Product Functions**

The main product functions are:

* Display graphs and charts to be viewed on the site
* Allow for easy interpretation of data for city planners

* 1. **User Classes and Characteristics**

The main users of our product are city planners: they are also the most important. These users will be viewing the data we analyzed to see where to allocate city resources. However our site will be public and open for anyone to access so we have kept our outlay simple for those everyday users that just like to view stats.

* 1. **Operating Environment**

The software will run on an apache server (v2.4.29) which runs php (v7.2.2), and mariaDB SQL database (v5.5.56) provided by the university. The php files will access our database to retrieve required data as specified from an AJAX request from the main site. The data will then be offloaded to the webpage, which will use javascript and the google charts library to generate our graphs and maps. Our site will load and display correctly on a web browser with a version greater than or equal to internet explorer v9, google chrome v55, firefox v59, and microsoft edge v40.

* 1. **Design and Implementation Constraints**

Due to client constraints, our software generates reports based off of a single snapshot of the mapping for change database. We do not have access to the mapping for change website and were given a .csv file that encapsulated a snapshot of all data deemed relevant by the client. We had to parse and clean the data we were given into a format that was queryable.

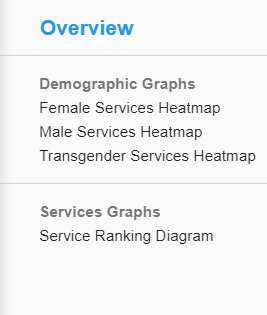
Our software should also support older versions of internet explorer (down to version 9). This constraint was brought to our attention due to the majority of city planners (one of our target demographics) utilizing older versions of this browser.

* 1. **User Documentation**

Our website is sufficiently simple that user documentation at this time is trivial. We have an index at the top of the page that allows a user to quickly jump down to the section they want to see, and the user can scroll. Visual elements are labeled. For information on how to use the google maps, see section 9 of the Software Design Document or consult the Software Users Manual.

* 1. **Assumptions and Dependencies**

We are assuming the data that we received is accurate and correct. Along with that we assume that the graphs we create will display useful information. Our system is dependent on the Google Charts library and Google Maps API; if Google crashes our charts will not display.

1. **External Interface Requirements**
   1. **User Interfaces**

Our interface is kept centralized to promote ease of access. Our goal was to display all information on one page, as per the client's request. There is a navigation sidebar which allows the user to easily navigate the page and jump to the graph they would like to view. The items displayed are all graphs of diagrams. The titles above each element give a brief description as to what the graph/diagram is displaying for faster and easier navigation.

* 1. **Hardware Interfaces**

Since our software does not have any designated hardware, it does not have any direct hardware interfaces. Our physical data is stored on a database, which is managed by the operating system on the server.

* 1. **Software Interfaces**

Our software runs off of an apache server v2.4.29, which allows us to use php v 7.2.2 and to generate SQL queries, which are executed on a MariaDB v5.5.56 SQL database. With those queries we retrieve the desired information off of the database in JSON format. Once our data is in JSON format, we use the google charts library to create a visual representation of our data and display that on our site.

* 1. **Communications Interfaces**

The user utilizes the AJAX interface when they connect to our webpage. Upon connection, the webpage sends off multiple AJAX requests to retrieve data from our database, first going through an apache php server then to the mariaDB SQL database. That data is the returned in a JSON string, which is then ran through the google charts library to create a visual that we display on our main page.

1. **System Features**
   1. **Display Graphs**

4.2.1 Description and Priority

This feature is of high priority as this is the main feature that the customer requested and all other features base themselves around this feature.

4.2.2 Stimulus/Response Sequences

From this page users can view all the graphs that we generate and navigate through them easily using the overview sidebar shown in section 3.1. As well some of the maps allow for user interaction so a user may zoom in and out to get a different perspective.

4.2.3 Functional Requirements

For this page to meet its functional requirements it must successfully load the graphs for the user to view. Aswell the navigation bar on the side to navigate through the graphs must take the user to the specified graph. The top navigation bar must also successfully take the user to the page they select.

Req1: Display graphs for the user.

Req2: Be able to navigate the page using the sidebar.

Req3: Allow the user to navigate to other pages using the top navigation bar.

1. **Other Nonfunctional Requirements**
   1. **Performance Requirements**

Since this is a real time system when requests are made from the user to access data our system should deliver the correct information in a reasonable time (we define reasonable time in this case as 2-3 seconds).

* 1. **Safety Requirements**

As the only service we provide is charts and maps, and it is fully public we have no safety issues that we need to accommodate.

* 1. **Security Requirements**

As there is no login to view data and all data is appropriate for public viewing, there are no security issues that we need to accommodate.

* 1. **Software Quality Attributes**

It is important that our software displays accurate correct graphs as these graphs will be used for city planning. Our software must also be highly accessible as it will be accessed through many different types of browsers. Our software should be highly usable, and robust to handle many users, and still display correct information. Our software should protect our database.

* 1. **Business Rules**

Since this is public software, everyone who accesses it has the same view, freedom of use, and functions available.

1. **Other Requirements**

Our database must be able to support expansion and scalability so that when new resources become available they can easily be added to the database.

**Appendix A: Glossary**

PHP REST API: A system that allows for dataset requests and deliveries. The requests are usually formulated as POST requests sent with an AJAX function.

AJAX: A JavaScript methodology that allows for background data requests utilizing the PHP REST API.

POST Request: A method for sending data to a server.