Software Requirement Specification

Project Vayu

Lab 2, Group 5

April 12, 2020

Oussama Saoudi, Lennon Yu
Christina Korsman, Diego Soriano

SFWRENG 2XB3
Software Engineering Practice and Experience:
Binding Theory to Practice
Department of Computing and Software
McMaster University

Contents

1 Introduction

1.1 Purpose

This document is intended to act as a software requirements specification for the program, which will outline the requirements that the software must meet and the priority of each requirement.

The intended audience for the document is the client requesting the software, who will use it to understand the formalized specification for the software project being developed. The software developers will also use the document in the development of components to ensure that they meet the requirements specified.

1.2 Scope

The software being developed is called Project: Vayu.

Project: Vayu aims to provide information on natural disasters, particularly information on property damage in sorted order, and geographical locations affected by particular natural disasters. The software will not add new data.

The objective of the project is to present natural disaster information in forms relevant to disaster relief organizations and governments. This will be done by presenting regions prone to certain disasters, what disasters they are prone to, which regions are most affected by natural disasters, and their property costs and estimated casualty count. Allowing for resource management of relief to be allocated to the right places.

1.3 Definitions, acronyms, and abbreviations

SRS - Software Requirement Specifications

1.4 References

[1] "Disaster Prediction App – Apps on Google Play," Google. [Online]. Available: `https://play.google.com/store/apps/details?id=com.disasterprediction.ios&hl=en_CA.` [Accessed: 07-Feb-2020].

[2] "MyFireWatch - Bushfire map information Australia", Myfirewatch.landgate.wa.gov.au, 2020. [Online]. Available: `https://myfirewatch.landgate.wa.gov.au/map.html.` [Accessed: 07- Feb- 2020].

1.5 Overview

The remainder of the document will consist of previous works and their purposes, the primary function of the software, user characteristics and the software's target audience, constraints on the software design, and finally assumptions and dependencies in the project. This document is formatted in standard SRS 1998 format.

2 Overall Description

2.1 Product perspective

Disaster Prediction App[1]

This mobile application reports space weather changes as well as earthquake incidents around the world. It shares similar characteristics with this project such as reporting recent weather trends (albeit on the interplanetary level) and this project as it is only concerned with solar influences and earthquakes, but not storms, blizzards, or tornados, etc., which are more relevant to daily lives. A way to improve it will be to include data reports on hazardous weather that are more often concerned by people, like those mentioned beforehand.

MyFireWatch[2]

MyFireWatch is a web service that visualizes recent fires by projecting their location on a map. Similar to the proposed project, it highlights regions most recently and often affected by natural hazards. However, it is limited to just visualizing incidences of fire, not analyzing if there is any relation between the incidences' geographic location and occurrence time, which may be implemented to enhance the service's usefulness.

2.2 System Interfaces

2.2.1 User Interfaces

The user interface will be a commandline interaction between the software and the user. By typing in the command with the repective parameter will allow the user to execute these operation.

Only 4 commands available, which sort, area, help , exit. Each command assits the user in operation of the program with mininal training. Their are failsafe if the user types in the command incorrect if such an event happens then the program will display a command help

Sort - sort the nodes by the weather type
Area - gives a area for the weather type
Help - give detail information.

Exit - Exit the program

2.2.2 Hardware Interfaces

No external hardware interfaces are required for this project.

2.2.3 Software Interfaces

Other than the standard Java project, this software does not interface with any other software products.

2.2.4 Communications Interfaces

This project does not interface with any communications systems.

2.2.5 Memory Constraints

The memory constraints is as much the user allows the software to have.

2.3 Product function

The main function that the software will perform is visualizing the danger of a certain geographical region. The geographical location will be provided by the user through manual input of positional coordinates.

2.4 User characteristics

The intended users are professionals in disaster relief, or other professionals who require detailed information on historic weather and disasters to either prepare for future disasters or deal with the aftermath.

2.5 Constraints

The reliability of the data is constrained, due to the non-infinity set of natural disasters recorded in the dataset. Another constraint with this data set is the limited set of data points relating to time. The format accepted for input data files shall only be of Comma Separated Value type.

3 Specific Features

3.1 Functional Requirements

[FR1.1] The system shall provide the user with a list of affected areas sorted by property damage cost.

[FR1.2] The system shall provide the user with a list of affected areas sorted by casualties

[FR1.3] The system shall provide the user with the list of affected locations sorted by proximity to a given location in latitude and longitude.

[FR2.1] The system shall output a set of convex hulls representing regions affected by a particular type of disaster.

[FR3.1] The output file shall be a text file

[FR4.1] The sorting of locations shall take no more than 15 minutes 95% of the time.

[FR4.2] User-submitted reviews on interface compatibility and ease of use must be at least 75% positive.

[FR4.3] The convex hulls representing disaster affected areas will cover at minimum 100% of the areas affected by the disaster type.The area unaffected by the disaster that may be covered by the convex will be at most 50% of the area of the convex hull.

3.2 Non-Functional Requirements

Scalability The system shall be able to handle input above what is required by 2XB3 course at McMaster University.

Reliability The system shall be tested with System Test Procedures to ensure it's reliability.

Accuracy of results The system shall be compared to otehr previous experiments with similar data to ensure the accuracy of results.

3.3 Design Constraints

The product design must be consistent with Mcmaster Software Engineering 2XB3 Software Engineering Practice and Experience: Binding Theory to Practice Final project documentation.

4 Development and Maintenance Process Requirements

4.1 System Test Procedures

4.1.1 Stress Test

A stress test consisting of 200,000 rows of data entries will be performed upon the finished software, and the software shall not crash or freeze during runtime unless external factors are involved in the stress test (e.g. power outage, physically interfering the hardware), or the performance of the hardware is below minimum requirement.

4.1.2 Correctness Test

The convex hulls representing disaster affected areas will be validated by checking that every disaster occurrence point of that type falls within a corresponding convex hull. If all points fall within a convex hull of their disaster type, then the software will be deemed correct.

4.1.3 Performance Test

A performance test consisting of recording the amount of time it takes to process the data for finding and creating the convex hulls, and amount of time it takes to sort the data.

4.2 Functional Requirements Priorities

- Sort Locations Based on Damage: Essential
- Output Affected Regions to File: Essential
- Display Affected Regions on Map: Conditional

4.3 Likely Changes to System Maintenance Procedure

Changes to system maintenance procedure may include but are not limited to the following:

- migration of the project to different version control providers.
- adding the use of issue trackers for bug fixing and functionality improvements.
- disallowing direct modification of the source code and only making changes by creating new files.

5 Software System Attributes

5.1 Maintainability

Using Git issue tracker to allow the ease of maintablity and process tracker. The product will use separation of concerns along with modularity to allow for ease of maintenance.

5.2 Portability

Through the use of java and the product will be allowed to run on any system that supports Java 8 and above.