Software Requirements Specification

Project: Threat Detect

Group 2: Harsh Patel, Michael Barreiros, Ben Li, Mustafa Choueib, Torja Istiaque

Made: Mar.1.19

Updated: April.12.19

Domain

Threat Detect is an application with the purpose to allow the users to identify, and remain vigilant of areas in the USA that are most common to having gun violence incidents. The user can get an overall outlook of which locations are considered safe determined by a numerical value within a scale that defines the level of danger. Stakeholders of Threat Detect include the developers working on the implementation. The individual developers that are a part of this group are Harsh Patel, Michael Barreiros, Ben Li, Mustafa Choueib, and Torja Istiaque. Government officials and citizens are also potential stakeholders. This application is useful to governments as it can provide specific information onto which locations to invest time and resources to increase public safety, revise gun laws, and continuously monitor the progress of their efforts. Citizens using Threat Detect are always aware of their safety, and the potential risks upon them when present in their specific location. Not only will Threat Detect assist in reducing gun violence in the USA, it will also identify common causes of this issue, and potentially allow the Government to implement solutions, such as help centres for mentally unstable people if that is a common factor. The future outlooks provided by Threat Detect is to ultimately keep gun violence to a minimum with actions taken by the Government, increase public safety, and reduce the number of casualties.

Functional Requirements

The program should:

1. Read and store incidents of gun violence from a file. The relevant parameters for each incident will contain city name, longitude/latitude, number of wounded, and number of casualties.
2. Take a location, determined by the user, as input and determine the region to which the location belongs.
3. Determine which of the incidents are found within a certain distance (which the user should also specify) of the location.
4. Calculate a severity index with a formula based number of killed and number of injured using the values of incidents within range of the desired location. Assign the threat level that corresponds to the calculated severity index based on the determined range of threat levels.
5. Display the list of incidents and threat level corresponding to the area around the desired location. Also, display the associated threat level.

Non-functional Requirements

Reliability: Given a location (address) as input the program should return the same incidents in a given area every time the program is run.

Robustness: The program should exhibit common error handling procedures, for instance the program should be able to handle incorrect user input such as an invalid address.

Performance: The program should be implemented such that it completes its tasks with accuracy, efficiency, and reasonable speed: a short response time and high throughput(rate of processing work). The program should use efficient sorting algorithms such as quicksort when sorting the array of incidents, and faster searching algorithms such as using Balanced Search Trees data structure to store and search through for a specific node for optimal performance and speed.

Usability: The programs client side code or the API should be easy to use. This can be done with descriptive method names.

Portability: The product will be implemented in a high-level language, Java. This allows the product to be easily accessible on different computers as the operating system is not an issue; Java is an OS independent language.

Maintainability: The product should be able to be updated after its delivery date. If any residual errors are present on initial release they should be able to be fixed and an updated version of the product should be pushed to users.

Timeliness: There should be a deliverable program by 11:59 on April 14th with all specifications, design report, project history in the form of a project log, functional application program, and the application code.

Visibility/Transparency: The development of the program should be visible. All the steps and the status of the program should be documented clearly. This allows the program to be worked on by multiple individuals at once. There should be no issues when an individual tries to implement their code into the current status of the program. The above will be achieved through the use of Git.

Scalability: The dataset that the program will use can be updated with more information and the program should be able to handle that change. As the dataset grows the program should be able to use the new information to give the user more information in the threat level and/or the number of incidents in the location they specify.

Requirements - Development & Maintenance Processes

To ensure the quality of the final product, rigorous testing will occur throughout the development process. The algorithms will be tested to ensure that they achieve reasonable runtimes. A possible maintenance consideration would be to update the product with newer data. A script could be created that pulls the newer data and inserts it into our existing data. Creating good documentation will also be key in the maintenance process as it will allow future changes such as transferring the application to work for another country to be easily achieved.