Group 5 Requirements Specification

Giacomo Loparco, Jacob Zarankin, Lucas Zacharewicz, Mohammad Hussain

The Domain

For this application, the domain is the group of entities who want the ability to view crime activity that they feel impacts their ability to live in a particular location. The goal fulfilled by developing this implementation is creating a streamlined process to eliminate potential properties in the city of Chicago based on the user's living style. The stakeholders for this project are people who want to purchase a house in Chicago in an area which would match their living conditions and where they would feel safe.

The main target audience are new home buyers, people with life changes that they would like to adapt to by moving to a new house/neighbourhood in Chicago. They would benefit from using the information presented through this application as it gives them a useful tool to search for houses/neighbourhoods in the city of Chicago that would best suit their particular lifestyle and/or living situation.

Functional Requirements

In the program, there will be user functions that deal with directly interacting with the user of the product. Inputs the user would enter are their address, personal crime weight rankings, or their life status (family, bachelor, elderly, etc.) for pre-made crime weight rankings already stored within the application. The corresponding output would be the ranking of neighbourhoods/areas in Chicago from best to worst based on their inputted specifications.

The application will also need to have functional methods, which do behind the scenes processing. The program would take the specified Chicago crime database information as its input. To create an output, it weighs each crime based on its severity and date occurred (more recent crimes are more relevant) and then sorts the crimes by this weighted crime index for each neighbourhood, to prepare the output for a given user.

Non-Functional Requirements

The program would focus on many non-functional requirements to ensure the best user experience. For the program to be reliable, the user should experience consistent results under the following conditions: they choose the same location, they hold the same weighted opinions, and they are using the same database. The user **SHOULD NOT** experience any significant bugs or issues that interfere with the functionality of the program. If the program were to fail, the user will see a detailed error report and the program will exit.

The accuracy of the application would be as high as the dataset would allow, but higher than a user would be able to get through passing through the data themselves, since the program would be able to accurately group and represent the data through a well-structured ranking mechanism for each crime, allowing for the user to make simple comparisons between many areas.

Performance is of high priority for this application, as it involves large amounts of data manipulation due to the massive database used. The user should not expect to wait more than 30 seconds for results to be accumulated, sorted, and displayed. Clever use of low cost algorithms will achieve this standard, and establish a consistently low runtime.

To deal with human-computer interface issues, the calculated data should be presented in a readable way to the user without any confusion. This can be achieved by making an easy to understand comparison system for the different crime indexes of different communities, to allow the user to easily interpret the information being produced and displayed to them, and therefor to easily make comparisons on multiple neighbourhoods at once.

Some operating constraints present in the project are: the database available for use, the processing power of the machine the program is running on, and the time complexity of each of algorithm used. All of these factors limit what the program can process and calculate. Some physical constraints present in the project are: balancing changes in crime, and how often the database used is updated. If there are certain neighbourhoods where crime is rapidly increasing or decreasing, if the database is not updated quickly enough to reflect this change, the program would be using old and inaccurate information.

The program will originally be made to run on desktop environments. Portability should not be an issue because Java runs on many types of environments, and a port could easily be created if need be (i.e mobile).

Requirements on the Development and Maintenance Process

To maintain quality control, the application can ask the user to see if the results given were accurate or were what they were looking for. If a user actually buys a house in that area, they can be prompted to write a user review of how the program helped them or how it could be improved. Also, the results can be compared to similar existing applications to see how similar they are or what areas of the program need work.

All of the functions doing calculations will be built in such a way that they are required to remain accurate and reliable as data is added over time. The speed should stay consistent (to some extent) even as the functionality of the program grows.

To improve the program, likely changes should be anticipated and accounted for. For example, the application can record the changes of crime rates in different neighbourhoods over time or account for old neighbourhoods being torn down/new neighbourhoods being built. If more databases are available, the program could expand into other cities as well instead of just Chicago.