Requirements Specifications

Group 03

CS 2XB3

**The Domain:**

**Stakeholders:**

* The first stakeholder for Habitat will be the development team as implementing this program correctly holds quite a large benefit for them
* People who are currently residents of Canada and are considering moving to a new location in Canada, are a major stakeholder
* Immigrants who are planning to move to Canada are a major stakeholder also
* Lastly real estate companies will also be a major stakeholder in the development of Habitat

**Stakeholder goals:**

* The development team’s goals are to ensure a quality product is developed that is easier for users of all experience levels to use. Additionally, making sure all algorithms implemented are efficient and quick is a major goal
* People who are moving within Canada need to be able to enter a list of locations and figure out in a quantitative method where the optimal place to move will be. Furthermore, they also need to be able to determine which qualities of life are most important to them and have the application catered to their needs
* People who are immigrating to Canada may not have prior knowledge about specific cities and so need to be able to just insert a province name and have a list of the best cities in that province. However, they will still also be able to input their own list of cities if they have done the research or have the prior knowledge
* Real estate companies are also going to be a heavy user of the product as it will allow them to quantitatively show their clients not just the value of a house, but the value of the neighborhood they are moving to

**Expectations of Habitat:**

* An easy to use UI that allows for the user to input a list of cities and have the top results given to them
* Fast and efficient calculation times for even large lists of cities
* The ability to present the quantitative data for quality of life in an easy to read and navigate output
* To give the user the option to prioritize which factors that are used to measure quality of life are most important to them
* A checker to see if there are any better alternatives within a close proximity to their best option

The main user base of Habitat will be people who need to move to a new location however lack the required technology to be able to measure the quality of life they will experience in that area. The above stated expectations, if accomplished, will ensure that this group of clients will have their needs satisfied in an efficient and easy to use manner.

**Functional Requirements:**

**Graphical User Interface** - Priority : Low

* Application should be accessible through a graphical user interface on top of a regular console interface

**Add and Edit Locations to the List** -Priority : High

* Application should allow for adding and editing locations to a list, which will be initially empty.
* Should allow for specify all cities with province(s).

**Specify Value of Factors that Influence Quality of Life Ranking** -Priority : High

* Application should allow for specifying preference of factors that influence the quality of life.
* E.g. high/low population, high/low temperatures, high/low income area.
* Factors include population, crime rate, climate and neighbourhood income.
* For each city, user specified values should be used as multipliers on each factors variance from the average of all the cities in the user specified list of cities.

**Prioritize Among Factors** - Priority : Low

* Application should allow for weighing some factors higher than others.
* Factors should also be able to be weighed as equal.

**Rank the List of Locations** - Priority : High

* Application needs to rank the user specified list of cities according to the user specified factors.

**View the Quality of Life Ranking Results in a List** - Priority : Medium

* Application should output of the application in a presentable manner.
* The output should be a list sorted best to worst locations for the user.

**User friendly Error Messages** - Priority : High

* If at any point the application needs to display a warning or error it should be understandable for the user.
* A error/warning should not cause the user to lose control over the application.

**No access to Internal Components** - Priority : High

* Application should not allow access to internal features, settings or any internal code.

**Check whether there is an alternative best option**- Priority : Medium

* The application should be able to check whether within a certain proximity to the user’s current best option, there exists a better place to live.

**Non - Functional Requirements:**

**Reliability:**

Software produced will either rank the 10 cities the user will input or return 10 cities that user would find appealing. This is based off of the weighting of factors that the user provides. By outputting the intended results and by consistently doing so (with robustness), our software meets its defined requirements and does what it is intended to do. No user data will be stored to promote security.

**Accuracy:**

Accuracy is relative for our project, as the true output depends on our decisions and algorithms to determine the quality of life. Habitat was created around these decisions to generate an accurate result to our definition of quality of life each time the program is run.

**Interoperability:**

Habitat will be a java-based application, so our goal is to ensure that our product will be compatible with as many computer operating systems as possible. It will be tested on multiple computers, including Windows, Mac, and Linux to ensure that no incompatibility issues exist.

**Maintainability:**

Each dataset has a specific access module to extract required data. This way, our code is modular and can be easily modified for future changes or potential extensions to our product.

**Performance:**

Our application is tested extensively to ensure it outputs a ranked list of cities to the users without a visible delay. Because the application is modular, we can easily add new performance optimizations to ensure the code can be improved further in the future.

**Robustness:**

Our program will behave reasonably in exceptional circumstances. If the user enters any input that our application can not accept, they are prompted to redo their input. This is done so that the application remains functional for the user without any disturbance.

**Requirements on the development and maintenance process:**

**Quality control:**

* JUnit for Testing
  + JUnit is the best framework for running test cases in Java
* Precise specifications for testing
  + Desired values and outputs we are expecting
* Thorough documentation will be required for testing code

**Priority of Functions:**

* The most important functionality which we need to prioritize is our searching and sorting algorithms (name searching algorithms)
  + The Binary Search will be implemented to search through our datasets
* The following priorities include data management and output:
  + This includes comparisons of averages (for data like crime rate or climate)

**Likely Changes:**

* Changes to our algorithms due to issues with its implementation. Possibly change the sorting algorithm entirely if the runtime is too high
* Change the range of cities. Could keep range to all of Canada or reduce it.
* We may sort the database prior to searching, if we decide that it is more efficient to do so, then we will implement it
  + System Maintenance
* Update database values if they are changed on their website
  + Development Process
* Job assignment might require some changes over the course of the project
* Clear documentation will be key for understanding each other's code

**Other Requirements:**

* Connecting our databases of Ontario to our program
* Debugging and testing
  + Separate Testing for modules
  + Map, interface, algorithm testing
  + Make any improvements to code and if there is enough time we could use the datasets of Canada
  + Maintenance requirements (data updates, UI updates, algorithm performance updates)
  + Plan for development, positions and potential future position changes
* Potential for expansion after due date to further develop
* Important note: plan early deadlines, if finished ahead of time can upscale