Requirements Specifcations Document

ReLocate

Madeeha Khan

Jenny Feng - Chen

Tasnim Noshin

Umme Salma Gardriwala

Patrick Laskowski

Domain

As the name suggests , Relocate helps new immigrants to Canada and other individuals looking for work fnd opportunities in various locations based on a large database of job outlooks and incomes for several positions . This application aims to narrow the demand and supply gap in the labour market , by directing labour towards regions where demand is greater . It also reduces the long and stressful process of job hunting and increases the economic well - being of the society.
Relocate answers the question: I am in this feld . Where are the best job and
income opportunities in Canada for me?

The primary stakeholders are individuals and families who are unemployed and are willing to move base in search of better work opportunities . Their goal is to move to a location where they are most likely to fnd work and expect to fnd employment in that location in a quick and easy fashion . By using Relocate , these families and individuals will fnd themselves in a better socioeconomic situation and support themselves through a stable source of income .

Functional Requirements

Upon entering the application, user should be greeted with a GUI asking them to input their search filters, the user must input a job name they are interested in searching for, but optionally they may input a province code (ON,BC, etc.) and a minimum average income for the city.

- If the user does not wish to filter by province code or income, they may leave the fields untouched to signify this
- Once the user has completed constructing their search query, they will press a button labelled “Search” to begin the search
- The program will read the datasets in and convert them to objects for easier access
- The program will search according to the user’s criteria and assign outlooks and income to each city
- The program will rank the cities according to their outlook
- The program will also find cities related to the current city via graphing algorithm

- When the search is complete, the results of the search will be printed out to the console. These results will consist of the city name, it's province, a numerical ranking for the job's outlook in this city, the reasoning for this outlook (pulled from the dataset), and the average median income of the city
- These results should also be written to a .txt file labelled "output.txt" located inside the data folder
- The user may decide to run another search query by repeating the process, or exit out of the GUI by pressing the X button to end the program

<u>Non - Functional Requirements</u>

Reliability

- The program must be reliable insofar as it needs to be .
- The program will be always be available to the user , as there is no plan for maintenance once the fnal product has been released .
- None of the searches or any of the personal information the user provides will be recorded or made available to any other party . This will ensure safety and security for the user , and will contribute to the integrity of the program .

Accuracy of results

- The program will be as accurate as possible , with the consideration that we do not have a natural language comprehension algorithm, so the user may not get everything they want with one search.

Performance

- The performance of the program will not fluctuate , since the database size is constant , and all of the searches will be performed in the same way .
- The speed of the program will not exceed one minute to generate the results for the user .

Human - computer interface issues

- The GUI will be very simple and always be functional (buttons will work as expected) and have appropriately labelled input felds .
- If a job is not found, the user will be informed.

Operating constraints

- The program will be able to run in tandem with other programs on the OS .
- The program will be able to run on Windows OS, Mac OS, and Linux OS .

Portability issues

- The program will only be able to run on platforms that can run Java, since it is a Java program.

Testing

- General
  - Check methods that are working properly before further implementation .
  - Test methods (sorting , search and graph)
  - Using Junit or other supporting tools
  - System and Integration Testing
  - Possibly testing the product in diferent environment dependencies
  - Features to be tested : Accept user input and product identifes it and do the proper job
  - test on GUI
  - s Generate proper formatted pdf fle with all the necessary information
  - Performance testing (how fast and how accurate)
  - Productivity and scalability

- Control Procedure
  - Report problem during the testing process , may need modifcation of the software

    - List the criteria for changes to the current product .

- Likely Changes
    - Algorithms (maybe using another is better)
    - User interface (GUI)
    - Multiple fle format
    - operating system changes
    - room to scale up for storage (updating data)

- Product Maintenance
    - Corrective : If the implementation is change , check whether the implementation corrects the error contained in the previous version of the product .
    - Adaptive : If changes on requirements in later production , the original specifcation must be adapted to the new requirements . Then implementation would need to be changed and back to check the correctness . (To keep the consistency , Update the specifcation before changing the implementation .)