System Requirements Specificaitions Documents

For JobSeeker

Prepared by Lab 03 Group 03

Zihao Du

Gengyun Wang

Senni Tan

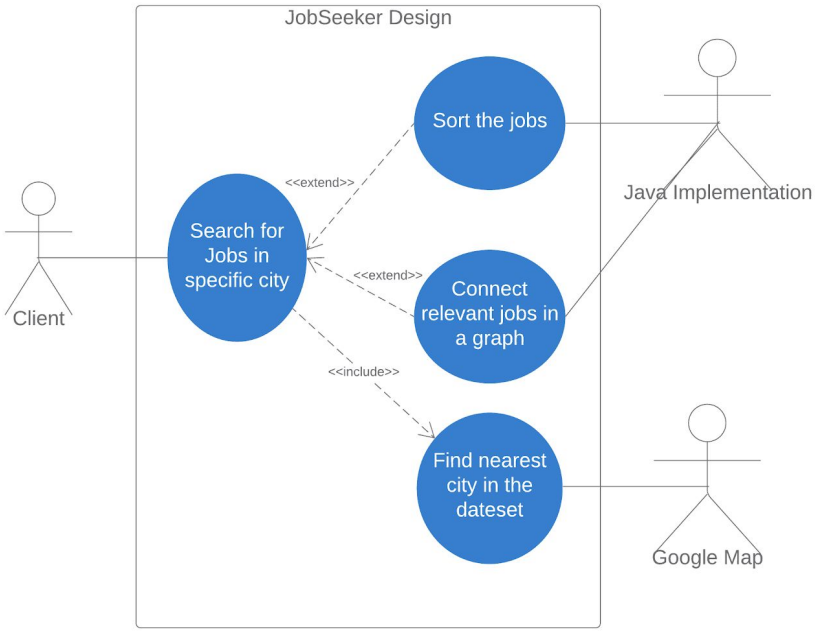
Wenzhi Wang

Domain

We will implement our project by using Java. This software will provide following services for several user types:

- **Students and Unemployed people**: This group of users usually are eager to find jobs with high potential as a good starting point of their career path. Therefore, after users select the job-field they want to search and their addresses, our software will give a list of satisfactory jobs from the adjacent areas, sorting by the job-prospect. Users can also type the NOC Code of the job they are interested to see the details.
- **People who want to change their job**: This group of users usually want to get a new job that relates to their previous working experience. Such users can find the relative works after they type the field and region they want to search.
- **Job market researcher:** This group of users usually want to analyze the performance or trend about a specific field or a specific job in a certain region and certain year. This goal is similar to the goal of the first group, but the researcher also wants to know past data, not only the latest data. Therefore we will set a year-input, the user can search the data in the period they type.

Functional requirements



Read Dataset Module

- This module will take the datasets as input and store the information into some data structure so that it can be used in later design.

- The input is in a CSV file and has over 100K rows, so the data structure should be able to contain these rows.

Searching Module

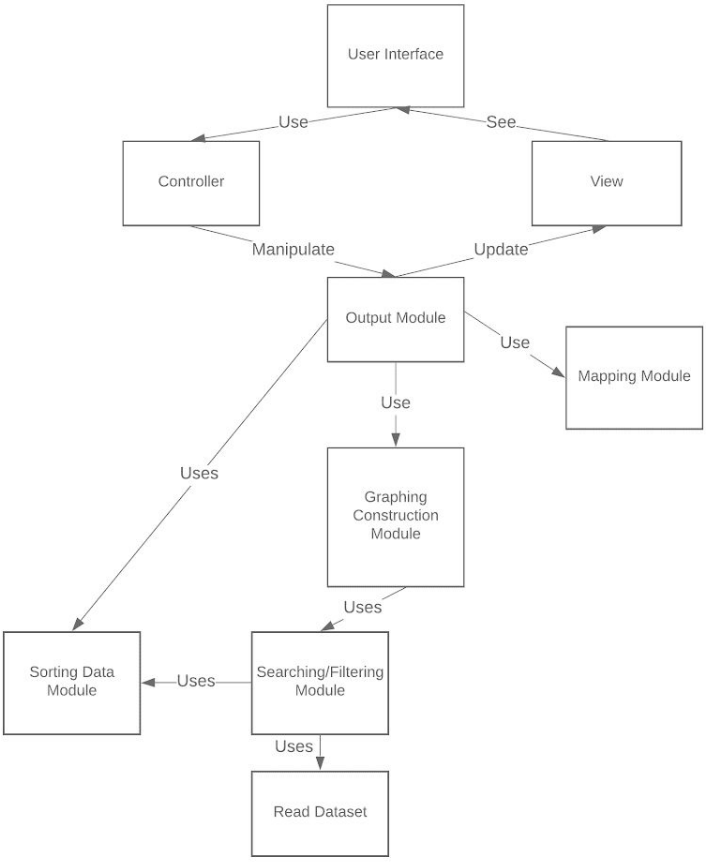
- This module searches for a special category of jobs and stores these jobs the client wants to see and store these jobs satisfying his/her requirements for further use.
- This module takes the data from the dataset as input and uses a sorting module before searching.

Sorting Module

- This module is used in the searching module to sort jobs with respect to numbers and also used in the graphing module to sort alphabetically.
- Need a changeable compareTo method to tackle different kinds of sorting problems.

Graphing Module

- This module takes the jobs(objects) as input, and constructs a graph. Relevant jobs are connected to each other.
- It will return the graph for other client modules.



Map Module

- This module uses a map API to calculate the

distance between cities, taking the user input city and returning the nearest city the dataset has.

Output Module

- This module takes the output of map module as input and constructs a graph of jobs which the client is interested in in the city.
- This module shows the list of the jobs(nodes) in the graph, and shows the number of relevant jobs. It also allows the client to have a close look at jobs in the list by showing job description, output and relevant job names.

User Interface

- This module takes the user's inputs and store them to control other modules
- This module also shows the user the output of the design and allows the user to update the inputs.

Nonfunctional Requirements

- Reliability --- The product should work with the satisfaction of design specification most of the time; constraints in reliability: the product will be tested by each group member and 99% of the test cases must be passed.
- Robustness --- The product would handle the unexpected inputs from the user; constraints for robustness: if the user gives the unexpected input, the input should not be taken and an error message will be displayed.
- Performance --- The program should run as fast as possible and the outputs should look generated; constraints in performance: the algorithm used should have a running time of at most nlogn, and the outputs should be displayed in a sorted alphabet order or by the order that the user selected in the filter.

- Usability --- The product should be easy to use for the user and there must be proper instructions that guides the user to use the product; constraints for usability: the prototype will be given to volunteers and the volunteers are invited to use the product, then the volunteers will be asked the question “do you think it is easy to use?”, at least 95% of the volunteers should say yes.
- Maintainability --- The product should be implemented in a way that makes the product modified after its initial release easy.
- Portability --- The product should satisfy that it can run in different environments, such as operating system; constraints: the product should work on the system of Windows, Linux and Mac OS.
- Understandability --- the requirements, design, implementation, documentation of software product, etc. should be easily understood.
- Accuracy of results --- the results for each input should be accurate; constraints: every output should be the relative information of the job that the user inputs, no information mismatch happened.
- Human-computer interface issues --- the result must be displayed on a screen.
- Physical constraints --- barrier: the user could only enter the string input that follows the instruction; if a user enters the string that does not match the instruction, the string will not be taken as an input and an error message will be displayed.
- Operating constraints --- if the user gives the input of category, the user can only select one of the categories on the list given.

Requirements on the development and maintenance process

Quality control procedures

- Multiple test cases would be chosen randomly to ensure each function works properly.

- Marginal input cases including non-existent job titles, invalid addresses, etc, should also be considered.
- During the testing process, each module will be tested by both the programmer and the tester.

Priorities of the required functions

- The searching, sorting, and graph modules are the core of the product and should be tested independently by each corresponding programmer. The input and output modules are of less priority in terms of the development process.

Likely changes to the system maintenance procedures

- In order to maintain the system, the team needs to keep its core functions correct and reliable, which involves keeping the data of job titles updated so that the client would not receive outdated or misleading information. The database, however, might not be updated as frequently as our system requires, and the maintenance could include checking the validity of the data set from other sources.
- If the user base expands in the future, it would be important to optimize the algorithms. It is possible that after long term usage, the team would be able to find more common user input cases and the resulting graph from such cases could be stored to improve the efficiency of our system.

Other requirements

- During the development process, a certain order based on the dependence of each module should be followed to increase productivity.
- For better quality control results, team members will review each member's code and avoid confusion.