System Specifications Document

JobViz
Software Engineering 2XB3
Group 5
L02

Rupinder Nagra, Jonathan Cels, Eshaan Chaudhari, Amir Afzali,
Jarrod Colwell

nagrar5@mcmaster.ca, celsj@mcmaster.ca, chaudhae@mcmaster.ca,
afzalia@mcmaster.ca, colwellj@mcmaster.ca

Table of Contents

Glossary

Purpose

Brief description of the project problem and solution.

Dependencies

Software and data that the project is reliant on that is outside of the scope of the project.

Domain

Brief description of the project's stakeholders, their expectations, and their goals.

Nonfunctional Requirements

Description of requirements specifying the quality attributes of the software. Includes the requirement name, description, and specification for completion.

Functional Requirements

Description of requirements specifying the function of the software system. Includes the required behaviour between input and output.

Development and Maintenance Requirements

Description of likely and unlikely changes to the projects, including how and why they would occur as well as a way to account for each change. Description of quality control, testing, and verification methods of the software development process.

Purpose

It is time consuming for current or prospective employees to analyze government data in order to find information regarding the job market. This project aims to give users useful and readable data through visualizations and intuitive data querying, in order to streamline job hunting and analysis.

The dataset used is the Ontario Sunshine List, which is a public sector salary disclosure of all Ontario employees with a total compensation of $100,000 yearly, or more.

Dependencies

The project is dependent on Unfolding Maps, a data visualization library, as well as the Government of Ontario, the resource manager controlling the Ontario Sunshine List.

Domain

A stakeholder of such a system are any roles that are actively involved in this project, and can affect or be affected by its outcome:

- The users or customers of the website, who are impacted by the quality of the website's service and value. It is important to manage their economic and social interests.
- Developers of the product, including the researchers, designers, programmers, testers and management
- Public sector employees whose information is being analyzed
- Project sponsors who are backing the project and the development team.
- The resource manager whose dataset is being used and analyzed.

The company must devote their attention to each of the objectives of a stakeholder, as every stakeholder in the system has different expectations from the project. These expectations are stated below:

- The user's goals are to be provided useful information regarding existing salaries and salary predictions for their desired field of work and location.
- The developers aim to deliver a useful and efficient product while keeping in mind concerns around safety or privacy. They need a well-defined design and specification in order for implementation to proceed properly.
- The public sector employees want to maintain their personal privacy and not have their data used for malicious or misleading purposes.
- The project sponsors overlook the execution of the project to assess if the product is feasible for customer use. The project must address a group of users' needs that can be formulated as a computational problem.
- The resource managers collect and control the data used in the project and want to ensure accuracy within the data.

Each stakeholders goals are different, and often these necessitate tradeoffs in the design and development of the project. The relationship between the expectations stated above are outlined below:

- The developers will need to keep in mind the expectations of the user for the information to be useful and predictive, while also ensuring that misleading information is not being spread about the public sector employees that could potentially affect decisions around their field of work.

- The project sponsors want to ensure that computational processing is done to the data, while the users want the product to be efficient and quick. This will affect the algorithmic and structural choices made by the developers.
- As the users and project sponsors both want the product to be useful, considering both perspectives of what that means for the project is necessary before specification design decisions are finalized.

Nonfunctional Requirements

Safety and Integrity Requirements:

Though the data of the public sector employees is publicly available, it is important that the software does not highlight any individuals outside of strict data-field comparison. Because the software intends to contain predictive measures, it is vital that misinformation around individuals is not spread.

Reliability and Availability Requirements:

Users deem the product useful if it shows them customized information quickly, and it can be used at their convenience. There should be little to no downtime (< 1 hour/week) where the product is unavailable. The data should be processed quickly and efficiently (< 5s to create visualization)

Utility Requirements:

The developers want to save both space and time while storing and processing the data. This will also have an effect on the algorithms and structures used within the implementation of the project.

Interfacing Requirements:

The UI should be easy to use as tested by a random selection of users with a varying range of technical background. Instructions and/or tutorialisation on use should be included. The users will be given a survey both before and after the tutorial with options ranging from 1 (difficult to use) to 5 (easy to use). The visualizations should be meaningful and obvious as to what they are showing. The same survey will have options ranging from 1 (unclear what the visual shows) to 5 (obvious what the visual shows).

Qualities:

For software to be considered good, it should display a range of important qualities. These qualities and their relation to the project are stated here:

Performance:

- The product should deliver the information to the user as fast as possible while also using as little energy and storage as possible. Poor performance often adversely affects the usability and, more importantly, scalability of a product so it is paramount that the product has high performance.

Correctness:

- The information provided to the user should satisfy the requirements outlined above, achieved through satisfying the stated requirements.

Robustness:

- The product should behave reasonably even in unanticipated or exceptional situations, achieved through satisfying the unstated requirements.

Verifiability:

- The product's 'Performance' and 'Correctness' should be easily measured and tested. Strong verifiability leads to the ability to improve the product much more efficiently as weak performance or incorrectness can be found.

Development and Maintenance Requirements

Quality control procedure:

The quality control procedure for the software product will involve a rigorous testing and verification process, followed by a period of regular use of the application, in order to identify issues that cannot be caught in traditional unit testing. This procedure should ensure that at launch time, the application is implemented correctly and carries desired behaviour.

Testing and verification:

The tester will ensure that the deliverable product undergoes sufficient testing and verification, in order to ensure correctness and a desirable level of usability. This testing will involve determining desired behaviour, such as that specified in the MIS, and testing in order to verify that the desired behaviour is achieved. The developers will perform the testing through unit testing frameworks. If each module is verified to be correct independently through carefully curated test cases, then the combined implementation of the modules can then also be assumed to be working correctly. The testing will involve verification of normal and edge scenarios.

Likely changes:

There are several aspects of the product that are likely to change throughout the software development process. The most prominent changes will likely be those to the user interface implementation, as well as available features in the application. While implementing, it may be determined that certain desired features, such as specific queries on the data set, or certain user interface configurations, may be infeasible or ineffective to develop. Given this, it is safe to assume that the available features of the application are subject to change from the time of design to the time of implementation. Another likely change is the exact sorting or searching methods that are to be implemented. The current design displays abstraction and leaves room for decisions on the algorithms that must be used. If these algorithms are determined to be infeasible given a certain scenario, then the algorithms are subject to change.

Unlikely changes:

There are several aspects of the product that are unlikely to change throughout the software development process. One such aspect includes the inflation rates that are to be used in the predictive salary algorithms. Standard inflation rates are to be pre determined and calculated. Given the fact that these inflation rates are an extraction of existing financial data, they will likely be constants that are not changing. Another aspect that is unlikely to change is the nature of the data set. The sunshine list is standardized, and there is no reason to assume that any manipulation will occur on the data set for parsing purposes.

Functional requirements

Motivation (Scope of the Work):

The motivation behind the product is the lack of compiled resources available to the public on jobs and the job markets. They are either very difficult to navigate or provide a narrow view of information. Users must have all the necessary information to make a sound choice when it comes to their career. From knowing which jobs pay more in a specific field to which companies have jobs in that field, every bit of information is important to this decision. The users of the product are any prospective employees looking for jobs in the public sector of Ontario. Due to the abundance of non-compiled information on public sector jobs, this information can be made more accessible to prospective employees.

When looking for jobs, factors that are taken into consideration include salary, location, company, salary growth rate, and field. JobViz will take the prospective employees input on what field of jobs they are looking for and provide an informative heat map that shows both which jobs pay more, and where they are located. This provides the basis for understanding the effects of the proposed changes being made to the process of prospective employees' looking for jobs in the public sector of Ontario.

Content (Context of the Work):

The system shall take inputs in the form of data requests, and provide outputs that contain filtered data corresponding to the input. These outputs are used to generate the application's UI and functionality. Inputs shall be checked for validity throughout the application's control flow, ensuring that each request to an interface or endpoint is coherent and applicable to determining the desired output.

The system shall have responses available in the case of abnormal input or circumstances, and fail-safe measures shall be in place in order to ensure that system failure does not occur. Abnormal cases shall be mitigated by following established communication pipelines between modules, as described in the application's module specification interface. Error handling and recovery procedures shall be in place for the application to be able to restore itself in these abnormal cases.

The system will evaluate inputs and determine the correct formulas and pipelines in order to return the desired output. Two primary outputs exist in the application: predicted salary, and list of salaries. A list of salaries output is achieved when the input is a sequence of filters that are to be applied to the data set. This output is achieved by performing algorithmic operations on the data, then aggregating and analysing subsections of the data. A predicted salary output is achieved when the input a profession contained in the data set. This output is achieved by determining the rate of growth for the profession as per the data set, and creating a predicted calculation of future salary.

References

Ghezzi, C., Jazayeri, M., & Mandrioli, D. (2003). *Fundamentals of software engineering*. Upper Saddle River, NJ: Prentice Hall.

Government of Ontario (2019). *Public sector salary disclosure.* Retrieved from https://www.ontario.ca/page/public-sector-salary-disclosure

Institute of Electrical and Electronics Engineers. (1998). *Ieee guide for developing system requirements specifications*. New York, NY.

Nagel. (n.d.). Unfolding Maps: Unfolding is a library to create interactive maps and geo-visualizations in Processing and Java. Retrieved from http://unfoldingmaps.org/

Volere Requirements Specification Template. (2007). Retrieved from https://www.volere.org/templates/volere-requirements-specification-template/