COMPSCI 2XB3:Computer Science Practice and Experience: Binding Theory to Practice
Project Proposal Template

Project Title:	JobViz
Lab Section Number:	**L02**
Student Names:	*Rupinder Nagra, Eshaan Chaudhari, Jonathan Cels, Amir Afzali, Jarrod Colwell*
Group Number:	**5**
Student McMaster Emails:	*nagrar5@mcmaster.ca, chaudhae@mcmaster.ca, celsj@mcmaster.ca, afzalia@mcmaster.ca, colwellj@mcmaster.ca*

By virtue of submitting this document I electronically sign and date that the work being submitted is my own individual work.

Abstract

It is unwieldy and time consuming for current or prospective employees to analyze government data in order to find the desired information. Our product aims to combat this by giving users useful and digestible data. This will be done through visualizations and intuitive querying, by showing the user data that they deem the most relevant. Users will be able to filter through visualizations relating to salaries and salary predictions to make decisions on work locations and fields.

The dataset our product uses is the *Ontario Sunshine List*, which is public sector salary disclosure on employees paid $100,000 yearly, or more. It contains information such as employee names, salaries, positions and employers.

For verification and validation, we will make use of JUnit, a unit testing framework built into Eclipse. This will allow us to make use of test-driven development and ensure we are on the right track every step of the way.

1. Objective

The project will provide current or prospective public sector employees in Ontario with information regarding existing salaries and salary predictions for their desired field of work and location.

COMPSCI 2XB3:Computer Science Practice and Experience: Binding Theory to Practice
Project Proposal Template

2. Motivation

The problem that motivates us to build this 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.

We believe that our users should have all the necessary information to make a sound choice when it comes to their career. Both 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 our 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, we believe we can make this information more accessible to these prospective employees.

When looking for jobs, some 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.

3. Prior Work

There are currently many websites that find salaries based on specific fields of work and locations. Examples of these websites are salary.com and payscale.com [1, 2]. A common pattern in the visual interface of these existing products is having input fields of a job title and a location, which after entering proceeds to tell you the average salary you might possibly receive in the specified area. Just like our implementation, these websites are also reliant on a location and salary database. However, our project plans to create a detailed visualization of the salaries, where users can find relationships between the various fields in the dataset. These fields are specifically the sectors they work in, salaries, employers, and job titles. This differentiates us from existing products that have a greater focus on finding and simply displaying an appropriate salary based on your job title and location. We plan on creating a detailed visualization using the UnfoldingMaps library in order for the user to have a more interactive experience with our project [3]. This is a library that can create detailed geo-visualizations and can be directly downloaded into Eclipse.

4. Input/output and proposed solutions

Dataset

COMPSCI 2XB3:Computer Science Practice and Experience: Binding Theory to Practice
Project Proposal Template

The following datasets will be combined to have a larger dataset that contains data over a couple of years to allow for more accurate predictions. This is why all datasets are needed.

2018 Public Sector Salary Dataset:

https://www.ontario.ca/page/public-sector-salary-disclosure-2018-all-sectors-and-seconded-employees

This dataset is the *Ontario Sunshine List* for 2018.

2017 Public Sector Salary Dataset:

https://www.ontario.ca/page/public-sector-salary-disclosure-2017-all-sectors-and-seconded-employees

This dataset is the *Ontario Sunshine List* for 2017.

2016 Public Sector Salary Dataset:

https://www.ontario.ca/page/public-sector-salary-disclosure-2016-all-sectors-and-seconded-employees

This dataset is the *Ontario Sunshine List* for 2016.

Output

Predicted salary: Predicted salary for a specific role/sector based on yearly trends from dataset

List of salaries: An extracted sequence of salaries from dataset for any specified sorting or priority sequence.

Solution

The proposed application takes various inputs and partitions the data set such that desired groupings are achieved. The extracted data is used as a basis for querying, visualization, and other aspects of the application's interface. Two fundamental outputs are described in this proposal, and a general scenario for each will be investigated. Algorithmic operations that are performed in the following descriptions will make use of the algorithms that are described in the next section of the report.

Suppose a user seeks to generate a map of Ontario, displaying an average salary in the healthcare industry for the year 2018. Assume that no data groups have been cached. First, the raw database will be sorted by publication year. Then, extraction of data from the year 2018 will be lexicographically sorted by the 'Sector' column. Following this, a search will be performed in order to extract all rows of data that pertain to the input sector of 'Healthcare'. At this stage, the data has been minimized to all instances of salaries that are from the year 2018, and in the health care sector. Now, the data is once again sorted, this time by the 'location' column. This final sort can be considered as the primary output. From here, one further manipulation is required. The available rows are iterated through, and average salaries are

determined for each location. Following this, all the necessary information is available in order to create the map. As can be seen in this description, a simple process of sorting and extraction can be used however many times as needed in order to output a final data set that conforms to the input requirements.

For the next scenario, suppose the user seeks to view a salary prediction for the University of Toronto Professors in the year 2020. Assume that no data groups have been cached. First, the raw database will be lexicographically sorted by the 'Job Title' column. Following this, a search will be performed in order to extract all rows of data that pertain to the input job of 'Professor'. From this, rows with the input employer of 'University of Toronto' will be extracted. At this stage, the data has been minimized to all instances of salaries that have the desired employer and job. The available rows are iterated through, and average salaries are determined for each year included in the data set. This could, for example, be 2017, 2018, 2019. From this, some growth percentage can be normalized and determined. This growth percentage is then used to create calculated prediction, which is the primary output in this case. As can be seen in this description, sorting and extraction, along with some basic calculations, can be used in order to generate predictions that conform to the input requirements.

All in all, it is evident that the available data can easily be manipulated in an organized manner in order to extract interesting and relevant information for the application.

5. Algorithmic challenges

Merge sort will be used as the sorting algorithm because the worst case time complexity is O(nlogn). This time complexity is better than other sorting algorithms such as insertion or quick sort whose time complexity is $O(n^2)$ [4]. The searching algorithm that will be used is binary search which has a time complexity of O(logn). The problem with using binary search frequently is that it helps find one element efficiently whereas this project will require finding multiple elements, and it will be inefficient. It will only be used for specific scenarios and efficient data structures will be used frequently for searching. In order to make consecutive searches efficient, hash tables will be used to organize the data and every row in the hash table will be a linked list for elements that have the same index. There will be multiple hash tables, and the hash functions will be determined by the property that needs to be searched for in the dataset such as searching by alphabetically or numerical values [5]. The shortest path algorithm will be used to determine the sectors that have similar salaries where the nodes will be sector and paths between nodes will be the average salary difference.

6. Project plan

Milestone	Deliverable	Due Date
Group Project Proposal Presentation	Project slides to present to the class	Week of Feb 10
Module Interface Specification	Create a Module Interface Specification based on the knowledge learned in 2AA4 course	Mar 7
Project Prototype	Intermediate prototype demonstration of the project	Week of March 9
Final Project Code	The visualization and Eclipse implementation of the project	Week of April 6
Final Project Presentation	Final project slides to present project to the class	Apr 12
Design Specifications	Document explaining the complete design process of the project	Apr 12
Peer Evaluation	Evaluation of the input of teammates over the duration of the project	Apr 12

References

[1] Salary.com, Available: https://www.salary.com/. [Accessed: 07- Feb- 2020].

[2] Payscale.com, Available: https://www.payscale.com/salary-calculator. [Accessed: 07- Feb- 2020].

[3] unfoldingmaps.org, Available: http://unfoldingmaps.org/. [Accessed: 07- Feb- 2020].

[4] studytonight.com, Available: https://www.studytonight.com/data-structures/bubble-sort/. [Accessed: 07- Feb- 2020].

[5] cs.cmu.edu, Available: https://www.cs.cmu.edu/~adamchik/15-121/lectures/Hashing/hashing.html. [Accessed: 07- Feb- 2020].