|  |  |
| --- | --- |
| **Project Title:** | **UniversitEATS** |
| **Lab Section Number:** | **L02** |
| **Student Names:** | *Smita Singh, Niyatha Rangarajan, Moksha Srinivasan, Jeffrey Wang, Jack Theriault* |
| **Group Number:** | **1** |
| **Student McMaster Emails:** | [*sings59@mcmaster.ca*](mailto:sings59@mcmaster.ca)*,* [*rangaran@mcmaster.ca*](mailto:rangaran@mcmaster.ca)*,* [*srinim1@mcmaster.ca*](mailto:srinim1@mcmaster.ca)*,* [*wangw115@mcmaster.ca*](mailto:wangw115@mcmaster.ca)*,* [*theriauj@mcmaster.ca*](mailto:theriauj@mcmaster.ca) |

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

Names and Signatures of Team Members

|  |  |
| --- | --- |
| Group Member 1 Name  **Smita Singh** | Group Member 1 Signature |
| Group Member 2 Name  **Niyatha Rangarajan** | Group Member 2 Signature |
| Group Member 3 Name  **Moksha Srinivasan** | Group Member 3 Signature |
| Group Member 4 Name  **Jeffrey Wang** | Group Member 4 Signature |
| Group Member 5 Name  **Jack Theriault** | Group Member 5 Signature |

## 

**Abstract**

Indecisive students spend too much time deciding what to eat while trying to stay within budget. This proposal outlines a project that seeks to solve this problem by streamlining the decision-making process. The project revolves around the use of the Yelp dataset, as well as a sorting algorithm which produces a set of objects based on user input. While similar products exist, this project is primarily targeted to university students, and will allow users to alter the sorting process to fit their preferences. To implement this, a number of challenges will need to be overcome in order to produce a product which can efficiently and consistently provide recommendations.

**1. Objective**

UniversitEats is a program targeted to university students which recommends cheap and highly-rated restaurants. Given an extensive data set, an algorithm will be implemented which weighs the various attributes of the restaurants, to select the best recommendation. As an addition, users will be able to rank their preferences to result in an even more personalized recommendation.

**2. Motivation**

We are motivated to build this project because young adults tend to struggle to find a quality place to eat nearby. Being young adults in the same position, this project is very dear to us. Usually, we are very indecisive trying to find restaurants , so an app that recommends a nearby food place according to what the user wants will be extremely valuable. Undoubtedly, most university students go out to eat and that can be time consuming, so finding an efficient way to facilitate this daily task is important for us. We expect university students to use this app, as a way to find a place to eat quickly so that they can focus more on their studies, and less on what to eat.

**3. Prior Work**

Several food services cater to the needs of their customers using their preferences. This is particularly used in applications like Uber Eats, SkipTheDishes, and DoorDash where food recommendations are made based on a customer's choices[1]. Using the choices, restaurants that match customer preferences and are within a certain proximity are displayed. Food preferences may involve types of cuisine: Mediterranean, Chinese, Indian, Mexican, etc.

Our product similarly involves taking in the user’s location and displaying a list of restaurants within a given proximity. However, the product we are building targets university students; making the choices more streamlined with respect to the university they study at. Thus, the key difference involves the range of options provided to the user. Being university-based narrows down the location to that of the area around the campus. Thus, the user can easily see which food of their choice is closest to their inputted location on campus.

Our product is unique in how we specifically target university students. The narrowed choices prevent the user from feeling overwhelmed, and helps the user make faster choices.

**4. Input/output and proposed solutions**

1.Yelp’s open dataset: <https://www.yelp.com/dataset>. This dataset is needed because it gives us all the information on nearby restaurants, their type of food, their price, and the amount of stars the restaurant has[2].

2. A restaurant: we will give the user the most optimal restaurant (concerning factors such as distance, rating, and type of food the user wants)

A refresh button: we will give the user a button to request another restaurant if they do not agree with the proposed restaurant (this will give the user the next most optimal restaurant)

Distance between inputted location and the restaurant: we will give the fastest route for them to get to the given restaurant.

3. The user will have decided by now that it is time to eat, so they will promptly open our app. At this point the user will input their location, food preference, if they care more about rating or distance, and the price range they want the food to be. After this, our sorting algorithm will go through every single restaurant that fits our needs (regarding distance, rating, and price) and then output the best restaurant that is nearby. After this, the user will be given a button if they would like to reroll and then if clicked, gives the next best option.

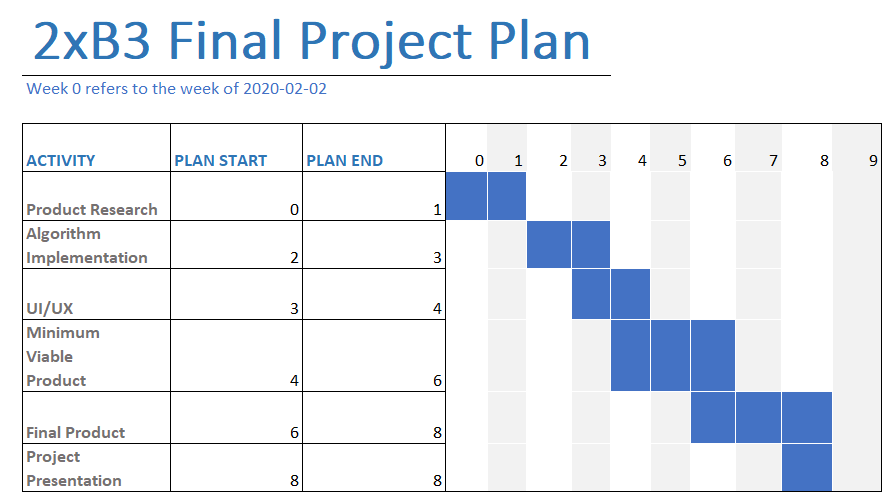
To decide what is the restaurant we will give, we will first discard all restaurants that do not fit the price range. After that, we will look if the user cares about distance or rating more. If they care about rating more, we will iterate the dataset and make a list of all the restaurants nearby (0-3km distance), if nothing with a rating over 3 is in that list, we expand the distance of restaurants (for example 3-10km). Once we have that list, we will then sort[3] the restaurants by rating. The highest rated restaurant is then given to the user. The refresh button can then be used to give the next rated restaurant. If the user cares more about distance, we will iterate the dataset but this time we will make a list of all highly-rated restaurants (3 stars +), if nothing fits then we expand the rating (2 stars +). Once we have this list, we will sort the restaurants by distance and give the closest restaurant. The refresh button gives the user the next item in that list.

**5. Algorithmic challenges:**

* We will require a sorting algorithm to do an overarching ranking and recommend a restaurant based on based on
  + The string/searching algorithm which parses data for certain keywords inputted by the user (i.e. Mexican, Italian)
  + The sorting algorithm that sorts restaurants by distance based on the location input
  + The sorting algorithm that sorts restaurants based on user ranking
* The challenges that we are currently aware of at this time:
  + Sorting/searching through a dataset of this size is difficult and time consuming, therefore it might require algorithms that are more difficult to implement.
  + Storing large data (main memory or externally)[4].
  + Optimizing speed of the searching and sorting.
  + Learning how to work with JSON data[5].

**6. Project plan**

*Include a table that describes at most 7 milestones and their deliverable for your project. Each milestone should have a specific date (e.g., Week 3).*



**References**

*You should have at least five references. Use a consistent format (IEEE, APA, etc.) for your references.*

Yelp.com. (2020). *Yelp Dataset*. [online] Available at: https://www.yelp.com/dataset [Accessed 7 Feb. 2020].

Power, L. (2020). *Food delivery apps in Toronto and how to order online*. [online] blogTO. Available at: https://www.blogto.com/eat\_drink/2014/01/food\_delivery\_in\_toronto\_and\_how\_to\_order\_online/ [Accessed 7 Feb. 2020].

Sedgewick, R. and Wayne, K. (2011). *Algorithms*. 4th ed. Boston, Massachusetts, United States: Addison-Wesley Professional.

Stack Overflow. (2020). *How to add element in List while iterating in java?*. [online] Available at: https://stackoverflow.com/questions/11177348/how-to-add-element-in-list-while-iterating-in-java/11177393 [Accessed 7 Feb. 2020].

W3schools.com. (2020). *JSON Introduction*. [online] Available at: https://www.w3schools.com/js/js\_json\_intro.asp [Accessed 7 Feb. 2020].

**Update: 2020-02-25**

* Updated Agile roles as recommended by the TA
* Update responsibilities
* Updated iteration plan

**Stakeholders:** TAs and Professor Samavi

**Team Leader:** Jeffrey Wang

**Development Team Members**: Smita Singh, Niyatha Rangarajan, Moksha Srinivasan, Jack Thierault

**Responsibility breakdown:**

|  |  |
| --- | --- |
| **Name** | **Responsibilities** |
| Jeffrey Wang | * Team Leader * JSON specialist |
| Smita Singh | * Development team member * Searching algorithm specialist |
| Jack Thierault | * Secretary * Development team member * Graphing algorithm specialist |
| Moksha Srinivasan | * Development team member * Front-end developer |
| Niyatha Rangarajan | * Development team member * Sorting algorithm specialist |

As an update to our previous project plan, we would like to highlight that we plan to have two iterations of our product. The first being the MVP, a working prototype of our product and then a final iteration with all features implemented and no glaring bugs.