CSE 403 Proposal Document
Amanda Loh (cosy16)
Amarpal Singh (amarpal)

Imagine that you are a student at UW, who has just gotten out of class at 5:30pm and you're hungry. You've texted all your friends and they're unavailable. You also have no idea what you want to eat. This is where our app comes into play. Our app enables you to choose what you want, when you want. With a simple Tinder-like interface, the app allows you to swipe "Yay" or "Nay" on different delicious food images sponsored from different businesses around the UW area. After a couple of swipe selections, an algorithm matches you to a restaurant and a person to dine with. If you just want to dine alone and use the app to decide on a restaurant, that option is also available. If you know where you want to eat but need a dining buddy, that's an option as well. Our app gives you the flexibility to find what you need for your impromptu dining experience.

The app is targeted primarily for UW students in the UW area. As the user group grows, as well as database info from restaurants from the Seattle area, the app could be later targeted for an audience outside of UW. This app is compelling because the addition of new features is easy. After the base app to match users with restaurants and food buddies is implemented, features such as filters can be added. Users can then filter for the gender of individuals they want to dine with to find restaurants, find food buddies, or both. Because we have the business aspect, where we allow businesses to input the pictures they want displayed in our app, our app has a unique twist.

This app is unique because competitors' products are either targeting a niche market or do not do real time pairing. For example, a competitor like Grouper are designed to arrange three friends with another group of three friends at a later date, whereas our app allows you to find someone to dine with right away. Other competitors like EatWith, HomeDine, and LeftoverSwap focus on travelers in foreign countries, attending dinner parties, and eating leftovers. Our app is about capitalizing on spontaneity.

The product will have a three-tier architecture. The main app will be done using Android or iOS and also a webpage for the business users to add pictures. The middle layer, or logic layer, will be a rest API written in either Java, Node.js, PHP, or Golang. And the last tier will be the persistent storage that will be queried by Java, Node.js, PHP, or Golang. There will be two login systems, one for businesses and one for end users. End users login will use Facebook API and use Facebook tokens for authentication, and authorization and business users will use the Yelp API. The Yelp API will be used to verify a valid business for the business side of the application.

The presentation layer, or front end, will communicate with the backend by exchanging JSON data. There will ideally be a few services that will exist and work together to provide all the functionality. There will also be a service that will allow businesses to login using Yelp API and allow the business to upload pictures of meals that are available and a product description. Additionally, the web page will display the current meals that have been uploaded and details

regarding those meals. The information collected from the web page will use the Yelp API and will populate the tables in the database.

With the information populated, users will then be able to login via Facebook and be presented with the user interface that will allow them to pair with other users. The user interface will allow users to pick how they want to dine, whether it is alone or with someone else, and whether they have a food preference or not. Depending on the mode chosen, the API will decide which JSON response will be sent after receiving each image. The rest API, after receiving the response from the user, will check whether there is a match or not and provide the appropriate response back. The mobile user will process the request and either continue swiping food or choose to be paired with the person selected by the app. Swiping will continue for a fixed time interval and if matches are not made, swiping time will be extended. To implement the matching, there will be a service that will populate the appropriate tables in the database and then read to see if a match has occurred. Additionally, a service will run at regular intervals that will clean the database tables.

In terms of technical ability, what makes this product unique is that it uses two popular services and adds a little of its own additional content to create a new application. The program has many interesting challenges. The program will heavily rely on database queries to organize the data and find matches and at the same time must maintain a clean database. The app is also unique because one set of users will use a web interface and the others will use a mobile app interface. Technical challenges with this app include multiple languages being used to write the two front ends and the back end of the application. The single most serious challenge we see in developing this product is coordinating all three components to work together to provide all features mentioned. There are essentially two front-ends, the rest API, and the back end. To mitigate or minimize the risk of not delivering on schedule, we will mock up the front ends and back end with dummy data and focus on implementing a smaller set of features at a time. To start, filters, such as choosing only restaurants, food buddies, and gender selection will not be implemented. If Facebook login is difficult, we may consider using Google or friends from contacts or hangouts instead. This will ensure we will have a working application that we can build upon without getting stuck. Initially, the main goal is just to get the matching aspect on the mobile application to work while first using dummy business data. If time permits, we will add the business side front end and back end so businesses can populate their own data. Then finally, filters will be added.