

Chiara Lim
Diego Sanchez
Kevin Qi
Marco Raigoza
Nathan Vollertsen

CS411 Final Project Deliverables

Table of Contents

Introduction	1
Deliverable 1: Proposal	2
Deliverable 2: User Stories	3
Deliverable 3: Prototyping API calls	6
Final Word: Evolution of the Idea	7
Appendix	8

Introduction

This document contains all of the deliverables from the semester: Proposal, User Stories, and API Prototype. Over the trajectory of the project, many changes were made to accommodate the challenges we faced from start to finish. We had made these changes visible throughout the documentation in **red** text to better highlight the timeline of adjustments when building the web application. The link to our Github Repository is seen below, containing source codes, prototypes, documentation and more:

<https://github.com/diegosbu/411-Software-Eng-Project>

Finally, we would like to direct your attention to this video presentation containing our project pitch, application installation, web-design walkthrough, limitations and adjustments, and future improvements. The link to the presentation is seen below:

<https://github.com/diegosbu/411-Software-Eng-Project/blob/main/docs/Application%20Project%20Presentation.mp4>

Deliverable 1: Proposal

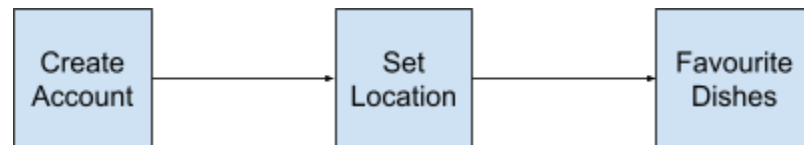
Our first and main project proposal was a food application that enables users to discover new places to dine at. Have you ever had trouble finding a restaurant to eat at? Maybe it's hard to decide between vague names like "Bob's Diner" and "The Grill". We want to create a web application that shows you a variety of generic dish images like "burgers" or "shrimp scampi". Swipe left and right till you've found your one true craving. Then, once you've decided, it provides a list of restaurant recommendations close to you that serve the dishes you want. It will utilize Yelp and BU Dining APIs to do so. Even better, try our Friend mode, which helps groups of friends decide where or what to eat by providing restaurants that have dishes that everyone likes.

Our second project proposal was a web application that combines the FBI crime data API, Airbnb API, and TripAdvisor API. The application will allow users to choose from hotels and Airbnb properties that are analyzed by their crime data in a given location. This ensures that users are able to know beforehand the safety of the neighbourhood they may be staying in. It also allows users to view various pricing options in a given neighbourhood. Safety and affordability will be at the fingertips of users.

Deliverable 2: User Stories

User Story 1 - Account Creation

As a new user, I want to make an account so that I can set my distance preference, keep track of my favourite restaurants (see *Favoriting Restaurants*), and keep track of my favourite dishes (see *Favoriting Dishes*).



Path:

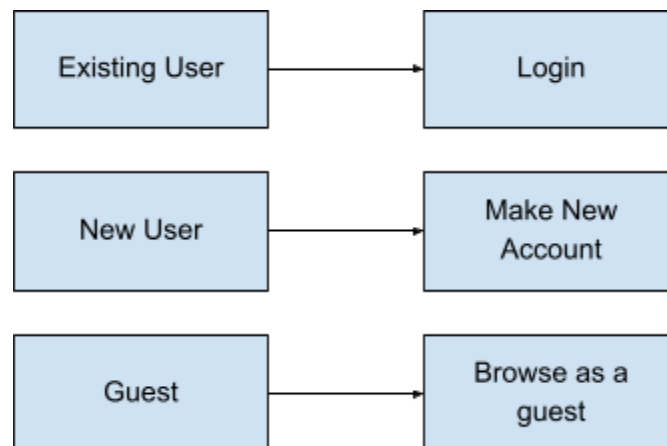
1. A user clicks on the account registration link.
2. The registration page loads up with username and password fields.
3. A user submits a form after filling in fields.
4. A user gets sent to the homepage as logged in user.

Changes:

- Users “create accounts” by logging in via their Google accounts.
- We were unable to implement a way for users to their distance preferences. Instead, users search for restaurants using the ‘Location’ search bar.

User Story 2 - Optional Login Flow

As a user, I want to have a choice to either make an account, log in to an existing account, or browse as a guest.



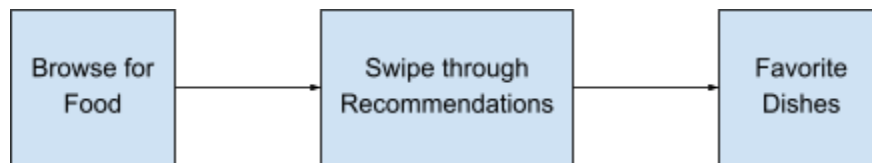
Path:

1. A user opens the application.
2. A user clicks on the account or guest login link. (Google Sign-in)
3. If login form clicked, page presented with username, password fields.

4. Once the form is submitted, the user is logged in and sent to the homepage as logged in user.
5. If guest login is clicked, the user is sent to the homepage as a guest user.

User Story 3 - Individual Browsing

As a user browsing for food options, I want to scroll through food recommendations so that I can find food that I want to eat. Then, I want the app to recommend restaurants that serve that food close to my location (ie. from restaurants in the area or from Boston University's dining halls).



Path:

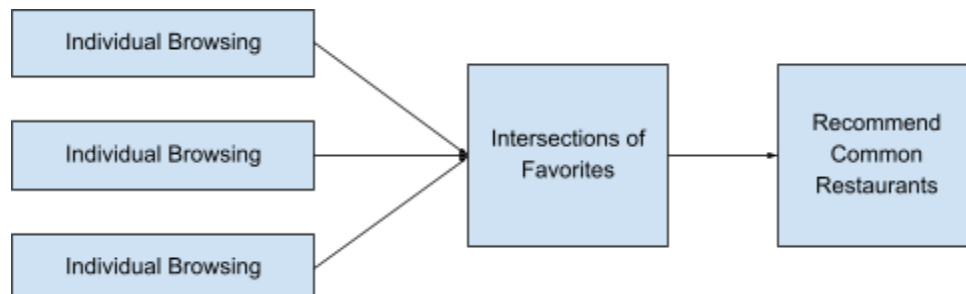
1. A user opens the application and completes the login process (see *Optional Login Flow*).
2. A user is presented with a series of food dishes and **scrolls (instead of swiping left or right)** to indicate interest.
3. Once a number of restaurants are found that serve dishes the user has an interest in, the application presents the list of restaurants to the user.

Changes:

- Instead of a user swiping through multiple dishes and then having the application generate a list of restaurants, the user is shown a generated list of restaurants based on location and/or restaurant name.

User Story 4 - Group Browsing [This user story no longer applies.]

As a group of users, we want to each go through the Individual Browsing workflow, and have the app find the intersections between what types of food we want to eat. Based on that, we want the app to recommend common restaurants that the group likes.



Path:

1. A user completes their individual browsing (see *Individual Browsing*).
2. The application finds intersecting dishes that users like.

3. The application recommends restaurants based on the intersections.

Changes:

- The current version of our application only allows single-user scrolling, and thus, multiple single-users/ group-use is not applicable.

User Story 5 - Favoriting Dishes [This user story no longer applies.]

As a logged-in user, I want to favourite some dishes so that they appear quickly while browsing to save some time.

Path:

1. A user is browsing food recommendations.
2. Favourites option appears on each presented food item.
3. A user clicks on the favourited option on the dish.
4. The favourite foods list is updated to reflect changes.

Changes:

- The current version of our application does not allow users to save their favourite dishes.

User Story 6 - Favoriting Restaurant [This user story no longer applies.]

As a logged-in user, I want to favourite some restaurants so that I may be able to access places that I enjoy eating at.

Path:

- A user gets a list of restaurant recommendations
- Favourites option appears on each restaurant in the list
- A user clicks on the favourites option.
- Favourite restaurants list updated to reflect the change.

Changes:

- Our current version of our application does not allow users to save their favourite restaurant.

Deliverable 3: Prototyping API calls

Language Decisions

For technology stacks, our group chose to go with Node.js. According to public data on GitHub pull requests, Javascript is and has been the most popular language on the site for at least a few years. Because of its dominating presence in the field, our team believes gaining more experience with Javascript in the backend would be very valuable. From an outside perspective, using the popular language is also helpful for readability and makes it more likely that a stranger would be able to understand our codebase.

Nonetheless, we did consider using Flask (Python) as our primary language. Again, according to the public data on GitHub pull requests, Python has consistently shown to be the second most popular language, and quite comparable to Javascript. Therefore for the sake of learning and codebase readability, Python and Javascript are rather similar. However, given that our front-end will likely be written in Javascript, it would be beneficial for us to have consistency in language between the two. By using Node.js, it increases visibility and eases the workflow.

Furthermore, we decided to use MongoDB as our framework for the web application. We feel that a NoSQL database such as MongoDB is very compatible with Node.js. It handles data in a JSON format, which is easily manipulated with Node.js. It is also generally more performant when compared to relational databases. We had also considered using MySQL. However, we find that for a small scale project, it is also easier to set up a data store that does not require normalization.

The prototype demonstration for this deliverable can be found on our Github repository under the filename "Prototype Demonstration.mp4".

Link:

<https://github.com/diegosbu/411-Software-Eng-Project/blob/main/docs/prototype/Prototype%20Demonstration.mp4>

Final Word: Evolution of the Idea

As we started the project and prototyping, it quickly became apparent that our initial API idea was unfeasible. The Yelp API we intended to use was deprecated and the new version was only available to businesses. Because of this, we ended up switching to Yelp's open-source API. Moreover, we switched from a menu-focused design to a restaurant-based design, due to Yelp's open-source API not including menu information. Using a combination of Yelp to find the genre of food and the location of the restaurant, we were able to provide most of the information used by our second API, which we use to grab food images matching the restaurant's genre of food. Along the way, we also found Google OAuth to be easier and more secure than dealing with our own account creation system. We also found the tactility required for a swiping to be beyond us at this time; however, we look forward to adding this functionality in future.

Appendix

Meeting Minutes [10/22/2021] - Initial User Stories

1. Login Page
 - a. New users
 - b. Returning users
2. Homepage
 - a. A “Start Search” redirects users to the Tinder Page to begin searching for food dishes to favourite.
 - b. Favourites Page - Favorite Food Tab & Favorite Restaurant Tab.
 - c. Account management Page.
3. Tinder Search Page
 - a. Food items show up for users to favourite and/or swipe.
 - b. Users can:
 - i. “Swipe” using keys and food items are used to generate recommendations [Swipe List]. After “swipe”, the next food item shows.
 - ii. At the same time, users can favourite food items using the star button [Favorite Food List].
 - c. Searching is done when the user exits/restarts/clicks done.
4. Recommendations Page
 - a. Shows summary of food items selected from tinder search [based on Swipe List].
 - b. Generated *restaurant* recommendations based on tinder search [based on Swipe List]. Users are able to favourite specific restaurant recommendations [Favorite Restaurant List].
5. Generic Restaurant Page
 - a. Gives menu, location, etc. information.
6. Favourite Food Tab
 - a. Manage Favorite Food List ie. remove, directly add
7. Favourite Restaurant Tab
 - a. Manage Favorite Restaurant List ie. remove, directly add
8. Account Management/ Setting Page
 - a. Set preferences for locations.
 - b. Set user preferences ie. username, password.