Two-Sided Recommendation Webapp Proposal

Joey Weiss, Jared Ranalli, Connor Johnson, Matthew Dodson

**Scope:**

The two-sided recommendation webapp (TSRW) is a recommender system that generates recommendations based on preferences from two different sides (e.g. a customer and a business). The TSRW will generate these recommendations in the form of coupons. Four different webapps will be built to collect preferences from both sides and then generate coupon recommendations. The first webapp is for both customers and businesses to build their profiles and submit their preferences, the second webapp is for a list of coupons to be recommended to the customer and for both sides to evaluate/rate the recommendations provided on a numeric scale (1-5), the third webapp is for third-party expert users to evaluate/rate the recommendations, and the fourth webapp is for users to act as the recommendation algorithm where they try to create matches better than the algorithm.

**Timeline/Goals:**

*(Tentative and Subject to Change)*

* Week 1 (2/14 - 2/20)
  + Submit project proposal and push to sponsors for feedback
  + Environment setup (Python, flask, etc)
  + Complete research and become familiar with React, Flask, and MySQL
  + Assign roles to group members
* Week 2 (2/21 - 2/27)
  + Frontend: Begin working on UI for webapp 1
  + Backend: Setup database integration with application for webapp 1 (Preferences from both “customers” and “businesses”)
* Week 3 (2/28 - 3/06)
  + Frontend: Finalize working demo of UI for webapp 1
  + Backend: Finalize database integration for storage of customer and business preferences
* Week 4 (3/07 - 3/13)
  + Frontend: Setup UI for webapp 2 for rating of the recommendations provided
  + Backend: Database integration for storing the ratings from both customer and business in webapp 2
* Weeks 5 and 6 (3/14 - 3/27)
  + Frontend: Finalize UI for webapp 2 and setup UI of webapp 3 for “expert” to login and perform ratings of the recommendations provided
  + Backend: Database integration for storing the expert ratings of the recommendation system in webapp 3
  + *3/19 - Midterm Presentation in class with* ***Functional Demo***
* Weeks 7 and 8 (3/28 - 4/10)
  + Frontend: Finalize UI for webapp 3 and setup UI of webapp 4 for users to act as the recommender algorithm and match customer and business preferences together
  + Backend: Setup database for webapp 4. Store user-generated “matches” to later compare against the matches generated by the recommender system
  + All: Begin to put together final paper
* Week 9 (4/11 - 4/17)
  + Frontend: Finalize and polish any bugs with the UI for the webapps
  + Backend: Finalize and polish any bugs with the webapps
  + *4/16 -* ***Final Papers Due***
* Week 10 (4/18 - 4/24)
  + *4/23 - Last day of undergraduate classes.* ***Final Project Presentation***

**Stretch Goals**

Adding a time tracker to the app which can track when a coupon will expire. When a coupon expires, the app will no longer give recommendations based on that coupon and will cancel all recommendations already given based on the coupon.

**Roles:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Frontend | Backend | Testers | Documentation |
| Joey Weiss |  | X |  | X |
| Matthew Dodson | X |  | X | X |
| Connor Johnson |  | X |  | X |
| Jared Ranalli | X |  | X | X |

**User Stories:**

1)

* Mary is a 20 year old college student who wants to get dinner. Her favorite food is pizza.
* Tony’s Pizzeria is trying to attract new customers from students at the local college. Tony has a coupon for half off a medium pizza.
* Mary makes an account with TSRW, specifies that she is a customer, and logs in.
* Mary opens up **Webapp 1** and is taken to the customer side. She specifies her age as 20 and her gender as female. She selects the “Food” Category. Underneath food, she selects Pizza as one of her preferences. She also selects a low-price range to be a preference for food as well.
* Tony logs into his business account with TSRW.
* Tony opens up **Webapp 1** and is taken to the business side. Tony’s specifies which coupons he would like to offer (We assume that the coupon for Tony’s Pizzeria is already known by TSRW.)
* Tony clicks “Add Items,” and lists the menu items he would like to sell. For every food item entered, he selects that it is a “Food” from a list of categories, and then answers what type of food it is from a list of food types. For example, Tony enters “Medium Pizza” as an item, says it is under the “Food” category, and of the type “Pizza.” Tony can also click “Add Preferences” to specify who he would like his coupons to be offered to. For example, he sets the preferred audience to be people aged 18 to 30. He also selects a preference for customers looking for food in a low-price range.
* The app sees that Mary and Tony’s preferences are shared. When Mary enters **Webapp 2**, she is given a recommendation for the coupon at Toni’s Pizzeria.
* Mary gets a medium pizza at Tony’s Pizzeria and enjoys it. She opens **Webapp 2** and gives the recommendation a 5-star rating.
* Chloe, a third-party expert, logs into TSRW, and opens **Webapp 3**. Chloe can see Mary’s account and the preferences she has. She can also see the account for Tony’s Pizzeria and the preferences of the business. Chloe can also see the list of all coupons known by TSRW. Chloe can see TSRW’s recommendation for Mary. Chloe also thinks that the recommendation was good and gives the recommendation 5 stars as an expert.

2)

* John wants to go see a movie. His favorite kind of movies are Sci-fi movies and Action movies.
* Worten’s movie theater is trying to sell more seats. They have a coupon for one free popcorn with any purchase of a ticket.
* John logs into his customer account with TSRW.
* John pens up the **Webapp 1** and is taken to the customer side. He goes to the movie category, and selects the genres of movies he prefers: “Sci-Fi” and “Action.” He also selects that he is looking for movies released in the current year, 2021. Additionally, he selects that he is looking for short movies (under 2 hours) which are in English.
* The owner of Worten’s, Larry, logs into his business account with TSRW.
* Larry opens up **Webapp 1 and** is taken to the business side. He selects the coupon he would like to offer (We assume that the coupon for Worten’s movie theater is already known by TSRW.)
* He clicks “Add Items,” and lists the movie he would like to sell tickets for. For every item entered, he selects that it is a “Movie” from a list of categories, and then answers what genre(s) it is from a list of genres. For example, Larry enters “Star Wars X” as an item, says it is under the “Movie” category, and of the genres “Action” and “Sci-Fi.” He also specifies that this movie was released in 2021, that the movie length as short (under 2 hours), and is in English.
* The app sees that John’s preference for movies matches the genres of “Star Wars X.” When John enters **Webapp 2**, he is given a list of coupons including the recommendation to use the coupon for a free popcorn at Warten’s.
* John does not like Warten’s movie theater, and so in **Webapp 2** he gives the recommendation a 1-star rating.
* Larry opens **Webapp 2** to see that his coupon was offered to people, including John. Since John fits the audience he preferred for his coupon, he gives the recommendation a 5-star rating.
* John then opens **Webapp 4**. He selects the category “Movies” and the genres “Sci-fi” and “Action.” When asked by the app to give a recommendation for this preference, he selects a coupon for “Tickets for Star Wars X: half off” at the IMAX movie theater – a recommendation he wished he got instead.

**Question Answers:**

1. How will you communicate?We will communicate through Discord, Email, and Zoom. The professor specified that he preferred to communicate through email, but we prefer to communicate regularly through Discord text chat. We plan to meet at least once a week through video chat on Discord.
2. What language will you use and why?

We plan to have a Python-based application because of its usefulness when dealing with data retrieval, manipulation, and analysis.

1. What frameworks did you use?

We plan to use the Flask and ReactJS

1. How will you test it?

We plan to test the application individually with local instances of Flask. We also plan to do automated Python testing using pytest.

1. What kind of architectural design will you use?

We were thinking of using a client-server architectural design. This may change once we have more familiarity with the external algorithm the client is using to create recommendations.

1. How will you be grouping the system into pieces of functionality or by subsystem?

Grouped into 4 separate webapps, each with a different function.

1. What additional tools do you think you need?

We will need to choose an SQL database management system. We are considering MySQL and PostgreSQL. Both should meet our requirements.

1. What do you foresee as possible issues?

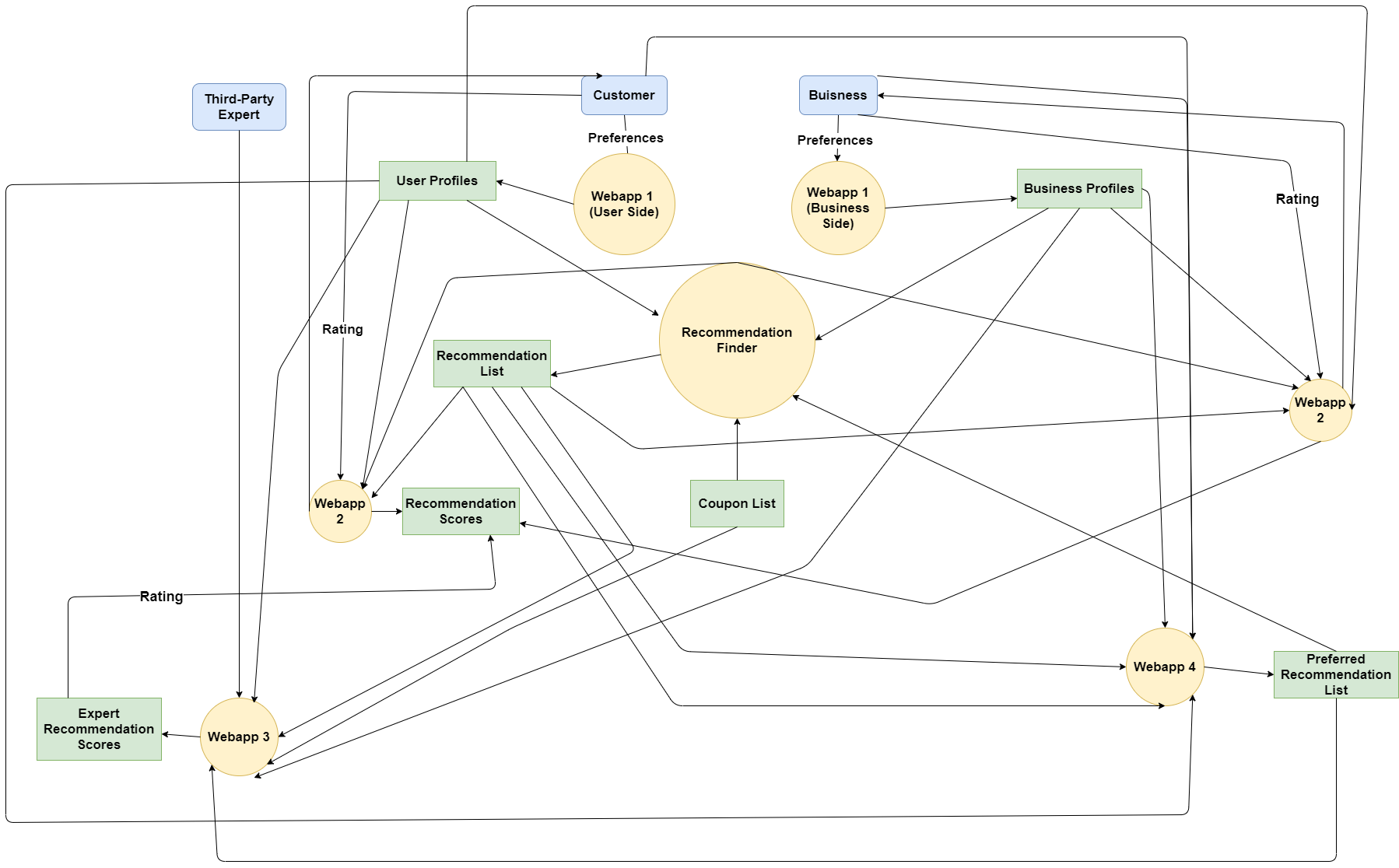
We expect that the front-end of the application may have issues because none of us specialize in front-end development. We hope that ReactJS will enable us to create aesthetically pleasing and functional webpages.

1. What systems will thus run on (OS X, Windows, Linux, etc)

Since this is a Python-based application, it should be able to run on any consumer desktop operating system (i.e. Windows, Mac OS, Linux, FreeBSD, etc.).

1. How often and how will you meet with the customer?

We plan to meet with them at least once a week. We may meet more times if we have particular roadblocks along the way.

**Data Flow Diagram:**