Two-Sided Recommendation Webapp Midterm Paper

Joey Weiss, Jared Ranalli, Connor Johnson, Matthew Dodson

Current Project Status

As of the halfway point in the semester, we have completed two of the four webapps for our project. The current plan of action is to first finish the functionality of all four webapps. Then, once we complete all the webapps, we would like to go back to make them more user intuitive. For that reason, we consider ourselves about half-way done with the project. We are confident that we will have it completed on time.

Course Adjustments and Change of Process

Since submitting the initial project proposal, we have removed a few steps from our project’s user stories. First of all, our project sponsor seemed to have no preference for how business coupons would be collected. Also, they did not provide us with any code for “the algorithm” which is meant to match the preferences between customers and businesses. For this reason, we were told by the sponsors to manually populate the database with coupons and recommendations. Therefore, we removed the steps regarding businesses inputting coupons and the webapp creating recommendations on its own.

We have also made a significant change to our “process” since the initial project proposal. One of the main requests from the sponsor was to put into place four separate webapps, each with a different functionality. After starting the project, we realized that having a single webapp that combines the functionalities of all four webapps into one is more self-intuitive. For example, the webapp could still limit what the different types of users see while also having less convoluted code. Thinking this was the best option, and with the assumption that the request for four webapps was done to compartmentalize functionality, we contacted the project sponsors to ask their thoughts on making the switch. After sending many emails over several weeks and receiving no response, which was unsurprising due to the lack of communication we’ve received thus far, we went ahead with making a single webapp with all of the requested functionality. If our project sponsors eventually reach out that they do still indeed want four separate webapps, we will take the necessary steps to change it.

The Software Development Process

Despite the lack of communication with our sponsors, communication within our team is working well. We have weekly meetings to establish tasks to complete before the next meeting and we use GitHub to collaborate and share code. We also utilize Discord when having difficulty with a task and to provide the team with status updates before passing off the work to another member. Due to our communication, we have been able to split development responsibilities based on individuals' areas of comfort. For example, team members more experienced with databases have taken over backend development, while those more experienced with HTML have taken over the frontend.

The biggest difficulty we’ve experienced with the software development process is getting our separate code to work together. Since we are mostly new to making webapps, we often complete a part of the code that fits within our skill set but then have difficulties getting the separate pieces of code to work together. For example, when we first had to output information from our database to an HTML page, not only did our backend developer create queries to get data and our frontend developer make a webpage to display it, but they also had to take a look at each others code and make edits in order to get everything to work together. While challenges like this are currently the greatest hurdle we’re having, we have found success in coming together as a team, regardless of job assignments, and fixing the errors together.

As mentioned earlier, another difficulty has been our inability to get in touch with our project sponsor. Numerous times we have had to wait several weeks before receiving an email response to our questions. With that said, we still feel confident in our ability to complete the project without further guidance. However, we lack the feedback necessary to know if the direction we are taking the project is accepted by our sponsors. To combat this, we are constantly pushing updates to the sponsor’s GitHub repository so that they always have updates to any progress that we make.

One of the positive things that we’ve experienced thus far was that the entire process of learning new technologies, getting started, and dividing up work was a lot easier than expected. Since most of our team was relatively new to developing webapps, we thought the learning curve would be pretty significant. As it turns out, the new technologies were picked up by our team quick enough to stay on track with the timeline initially outlined in the project proposal.

Project Risks

The biggest risk for the project would be if the sponsors added more requests to the project requirements. The project seems to function as a framework to collect and output data for what they have called “the algorithm,” but we have received a vague description of what that is. If the algorithm has already been completed, we do not know if our webapp will need to change to work with it. If the algorithm has not been completed, we assume it will be written to work with the project we create. Given the short amount of time left, if our sponsor requests changes too close to the deadline, we may not be able to finish them on time. That being said, we believe that outcome is unlikely and that our project will meet all of the sponsor’s expectations.

Trade-offs and Priority

One of the major trade-offs that we decided to make as a team from the start is the sacrifice of UX/UI design for functionality. In order to ensure we meet all requirements set by our sponsors, we decided the best approach would be to get all functionality in place before doubling back to improve the user interface of the webapps. Another reason we decided this is because the main use of our webapp will be for research purposes. Therefore, with a very limited number of users, we determined a higher priority needs to be placed on functionality rather than design. However, as long as there are no major setbacks to our timeline, we have built in an adequate amount of time following the completion of webapp 4 to go back and improve the front-end design and UI of all four webapps.

Future Plans

For the rest of the term, we plan on continuing our functionality-first approach and completing webapp 3 and 4 before going back to make other lower-priority improvements. With most of the groundwork completed for the webapps, we are on track to completing webapp 4 a week or two before the end of the term. This will give the team enough time to go back and polish up the user interface of the four webapps.

Database Schema

As seen below in figure 1, our web apps are using a total of eight tables thus far. We have completed webapp 1, which implemented a login system to populate the user table with a primary user id, a unique email, a password, a name, and their user type of customer or business. Webapp 1 also allowed for the user, either customer or business, to set up their profile and set their preferences. By setting up their profile, the businesses/customers table was populated in the database. Then, when setting up preferences, the customer supplied information to populate the customer\_movie\_preferences table and the business supplied information to populate the business\_movie\_item table. We have also completed webapp 2. This webapp recommended coupons from the coupon\_list table to the current customer/business. Upon rating the coupon recommendation provided, the recommendation\_list table is updated with the customer/business score. In webapp 3, we will add an additional user type of expert where that user can rate the recommendations made to both the customers and businesses. Upon rating the recommendations, the expert\_score field of the recommendation\_list table will be updated. Then, in webapp 4, we will have the users match coupons to customers in the attempt to out-perform the recommendation algorithm. As of now, we have not yet implemented the functionality for this.

Figure 1:

