Final Report - Team 034 - CursedQueries
Jeffrey Wong, Min Jae Kim, Will Nelson, Sneha Sundar
(jwong19, minjaek811, wbn3, snehas9)

- Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).

Overall, the project did not deviate much from its original design. Other than having a different UI, our project has a page for login, search, my groups, and profile. We decided to simplify our groups page to basic creation/deletion/edit functions. Additionally, we do not have the "more group info" page embedded in our search page, which we want to implement in the future.

- Discuss what you think your application achieved or failed to achieve regarding its usefulness.

In the current standpoint, our application failed to achieve its usefulness. Currently, we only have a functional login, search, and profile page. We still need to get our group page working which manages group creation/deletion for the application's core functionality. For a better user experience, we should also have a page to view group information (e.g. members in the group, location) and have more complex search filters in our search page.

- Discuss if you changed the schema or source of the data for your application

We have not changed the schema or data sources from stage 3. In the future, we want to remove the data manually inputted by the user (users and studygroups) in order to add a sequencer for primary keys when making new users/groups.

- Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?

There are no changes to the ER diagram/table implementations, but in the future we want to remove excess columns that we don't use in the database (e.g. group_size from StudyGroup as we could just get that from the Membership table)

- Discuss what functionalities you added or removed. Why?

We added a login page to differentiate between users. We simplified the application's functionalities to its bare components: homepage, search, profile, group manager page. Due to time constraints, our group manager page still does not work and we have yet to finish our group manager page and test our matchmaking for our creative component, which involves executing a transaction to match study groups based on available meeting times. We also removed the request to join a group, which should be implemented in the future for the group manager to handle.

- Explain how you think your advanced database programs complement your application.

The advanced database programs are best suited for the search page, as we could apply complex filters to our search which makes it suitable when there are a lot of groups inserted into the application. In the future, we could have queries/transactions to have a dynamic handler for adding/editing groups/contacts depending on if they already exist or not in the database.

- Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.

Jeffrey: The backend will keep on denying frontend requests until CORS authorization is applied for the frontend. CORS allows foreign websites (in this case, the localhost from the frontend) to communicate with the backend. Additionally for the backend, "GET" and "POST" routes may need different methods to extract parameters passed from the frontend ("GET": uses request.args.get() and may need a dictionary setting to be passed from the frontend, "POST": you could just get the parameters from a json version of the request's args). Do it the wrong way, and the request doesn't extract the args or just fails outright.

Min Jae: The most challenging part was the debugging process with api call malfunctions. I had spent a lot of time checking the connectivity between the backend and the frontend, but initially neither the frontend console logs nor the backend logs provided clear information of what was going wrong. It turned out to be an issue with URLSearchParams, the backend did not interpret the query parameters correctly, leading to empty or failed responses.

Sneha: One technical challenge I experienced was getting data from the backend to the frontend. This became especially challenging with the user_id field from the users table in the database. We needed this data on the frontend so that we could have that information for the login and create account features. One piece of advice that would have been helpful here is to double check that your column names in the database match how you're indexing your data in your backend call. The other thing that was super helpful with creating a global state on the frontend to keep track of this information was React's useContext which we needed to share state because user_id was a parameter that we needed for multiple components.

Will: I had issues getting the user's groups to display as options in a drop down list. I ended up using the groups.map function to get it to work:
groups.map((group) => <option value =
\{group.group_id}>{group.group_id}</option>).

- Are there other things that changed comparing the final application with the original proposal?

None, our fundamentals on the application layout is the same other than simplifying the search, group manager, and profile.

- Describe future work that you think, other than the interface, that the application can improve on

There could be a lot of work in our database implementation that would make our backend processing more convenient. For example, we want to delete the existing data in users and studygroups so we could add a sequencer for their primary keys for future insertions into the tables by the user. We should also integrate the queries we have in stage 3 to our search page to have a more dynamic search. We also should apply transactions to the database for our backend route handlers to call whenever we execute more than one query there.

- Describe the final division of labor and how well you managed teamwork.

The division of labor was mostly equal, with each member handling one page for both the backend/frontend. If anything, there should have been more meetings right after checkpoint 3 so progress wouldn't lag behind, which probably would have allowed us to finish the project.