CS 411 Final Report

- **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).**

- We missed the user inputting their ideal couple neighborhoods they were interested in.
- We missed the map that would show the closest matches from what they inputted.
- We did not implement the user to keep searching by putting in new metrics, only budget was allowed.

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

Our project succeeded in developing an apartment search application aimed at assisting tenants in finding suitable rental properties in Chicago based on price and crime rates in the area. We made it possible for users to filter and prioritize available apartments accordingly. From this they can click and inspect neighborhoods for their crime statistics as well as price and make an informed decision on their place of living. However, we did fail in creating a login and unique userID preference system. This makes it more difficult to keep track of previously inspected apartments.

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

The sources of our data changed quite a bit from the proposal, although with the crime dataset there were tens of thousands of entries with many data fields, which we thus heavily filtered down to accommodate our SQL tables. However, for neighborhood, apartments, users, and preferences data we generated entries using AI and manually entered them into our own tables to work with our application.

- **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 a few changes from our UML/ER diagram from stage 2 in comparison to our final product which reflect us streamlining which datasets were necessary for the user at the end of the day. For one, we ultimately chose to omit the RenterCompany dataset in our final design. This was because finding sources for renter company data in general proved to be very difficult, and also the information provided from that dataset did not meaningfully connect to our other datasets as we began moving forward with generated data as well. We ended up replacing this

with an Apartments Table, and just added a RenterCompanyName attribute. This fit better with our overall application and connected with the Neighborhoods table better. Additionally, we added a Preferences table to our schema. This dataset did meaningfully link with the User data and proved to be a core component in our final product, and was more suitable in our design than our original vision that had preferences only as a relationship between users and neighborhoods. Lastly, we combined the crime type with the crime event table. As we began creating the database in stage 3, this made the most sense primary-key-wise and to connect it with the neighborhoods table.

- **Discuss what functionalities you added or removed. Why?**

One of the primary functions that was ultimately removed was overlaying our apartment information on a map of Chicago, with a color metric showing which apartments were closest to budget and which were farther away. This functionality was removed due to conflicts with our neighborhood dataset making the map format very difficult to read for a user. Specifically, in our neighborhood dataset there were certain neighborhoods that were subsections of others, or a larger general area that encompassed multiple areas. As such, there were lots of overlapping points and readability was lost, so we ultimately went with a list format instead, with a user giving their initial budget estimate.

We added to our preferences functionality from our initial proposal as it gave the user a better way to filter out neighborhoods they were not interested in whatsoever. However, we were unable to make it functional with a unique user ID, due to not having a login page. Though, we did do the initial setup and frontend interface to display preferences.

We ended up removing filtering by nearby attractions and other metrics. This became too difficult to complete during our time frame.

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

The dataset of the chicago crime set provided by the city of chicago gave us 22 columns that we condensed down to Date, Location, Primary Type of crime, Description, and whether an arrest was made or not. This allowed us to get our statistics for the application when searching the neighborhoods, which gave the user extra info to make an informed decision about an apartment.

- **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.

Andrew: One technical challenge that I encountered was navigating through the GCP and utilizing it to test the advanced queries and indexing. Google Cloud Platform was not something I had used in previous classes and was a new experience for me. This made it a slower experience for me compared to someone who has used it often. I found the workshops very helpful to getting me up to speed on this platform and I highly recommend it to future students. Rewatching the videos posted on the workshops are helpful, but attending in person was the most helpful for me, especially if you have never used GCP before like me.

Joe: I had very little web development experience before this project, especially with back end frameworks and libraries. I found it difficult navigating the initial code base and seeing how the front end navigates with the back end. It was eye opening seeing how the server acts as the middle man with the client and database, considering I am also taking CS341. I definitely got a lot better at debugging throughout the course of this project, specifically using "Inspect Element" and the console.log() call. These specifically came in handy when I was trying to run the stored procedure in the back end. It made me realize I had an empty JSON, or I was parsing the response incorrectly. Ultimately, this project made me much more familiar and confident working with full stack projects.

Srijan: The most difficult aspect for me in this project was navigating and doing work between the frontend and backend for our final product in GCP. Coming in I had some familiarity with HTML/CSS frontend development, but I was unfamiliar with using NodeJS for the backend. While the guides provided by CS 411 staff proved helpful in setting up a template for our frontend and backend, actually modifying it to accommodate our project was something I had to get accustomed to over time. Variable names not matching up, datasets showing up as null, and having aspects of UI not show up on the frontend were some problems that continuously showed up during development, and addressing them required a fair amount of supplemental research. Ultimately however, I came out definitely more seasoned with using GCP and NodeJS as a result.

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

The big thing was scrapping the map functionality and instead trying to add preferences functionality. We also only added functionality to filter by budget, not by other factors like crime and other neighborhood attributes. We did, however, provide users with accurate information on

crime statistics after clicking on that neighborhood. Ultimately, our final application wasn't as visibly intuitive as we hoped, and we lost some functionality.

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

If we improved on another analysis besides the initial budget would make the application more versatile to have more parameters to narrow down the search even further. Our website to see the opening boxes asking their budget for house hunting but ran out of time on inputting their ideal couple neighborhoods they were interested in. Adding that would allow more personal results. We can improve on allowing the user to see nearby neighborhoods to where the location they entered and see if a different nearby neighborhood is a better match for them or not. A map with green markers representing ideal matches and matches with fewer preferences met being in red would allow for an easier and quicker experience to find neighborhoods. Additionally, embedding a link to the actual apartment could make the search even more robust and easier.

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

For a final division of labor, work was divided evenly between all members. For collecting datasets, Andrew gathered some sets on Chicago's crime rate. Additionally, some information which couldn't be exported into a dataset was generated and input manually as SQL tables by all 3 of us. Srijan and Joe were responsible for the front end UI work. The backend systems were done using NodeJS by Joe and Srijan with the datasets provided by Andrew, with all work being periodically reviewed by all 3 of us. Finally, any pushes to Github/Prairielearn and any other changes on Github were handled by Andrew as he is the group manager. We worked well together as a team, asking for help when needed, and everyone carried their weight.