Team 062-Prequel Final Report
EAT@ILLINOIS

Final Demo video link: https://www.youtube.com/watch?v=pw97Jtv78jo

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

   We maintained most of the original functionality from our proposal, with some minor differences. We did not end up adding functionality to filter by opening/closing time as it was hard to find accurate information for many restaurants, so it would be pointless. The layout mockup ended up very different from the proposal, as we had some difficulties as well as time constraints with implementing a more advanced UI using JavaScript React. Otherwise, the application was mostly consistent with the initial proposal.

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

   We believe that the database achieves its purpose of recording restaurants and reviews pretty well and the advanced queries, triggers, and procedures we made were conducive to a useful application that fit with our initial vision for the project. All of them were relevant to what we felt were common questions that would be asked by an actual user confronting these situations. However, our application did not have a proper user login to automatically record who was writing reviews, which would have improved usability significantly. Also, we were unable to achieve the usefulness goal of opening/closing times due to inconsistent data sources.

3. Discuss if you change the schema or source of the data for your application.

   The customers table was modified from the initial plans to include a column to record a customer's credibility based on the number of reviews they had previously made. We also considered adding the auto-increment tag to various ID columns such as restaurantID. We did not end up changing the data source for the restaurants (Google Maps), but as an

improvement we could potentially use Yelp because it has cuisine information and more details on opening hours.

4. 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?

   We did not make any changes between the relationships of the tables from the initial design in this aspect. Our initial design proved suitable for all the functionality we wanted, and it made logical sense. So, we did not end up changing the design (besides the small change to the Customers schema). However, we did consider adding auto incrementing indices, but this would have required deleting all the data and re-adding it, which we did not desire to do in case of any problems before the final demo.

5. Discuss what functionalities you added or removed. Why?

   We removed the user-login authentication as it added an unnecessary amount of complexity to our application and just added a frontend user-login interface.
   We also removed the sorting functionality from the original plan and instead added a more useful functionality that filters popular dishes based on either the number of purchases or average ratings from customer reviews.
   The Google Maps creative feature was omitted due to lack of time, and the lack of opening and closing times was omitted due to lack of useful data available.

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

   Our advanced database program strengthens our application's overall usefulness by giving customers (reviewers) an individualized experience. Customers can add their purchases' reviews and by doing this improve their credibility level. The added reviews also insert new purchase information in the Purchases table. Our stored procedure then evaluates a couple of complex queries that use Purchases and Reviews table to find highest sold dishes within restaurants and dishes with GOOD average ratings respectively, which helps to pick out dishes most preferred by customers.

7. 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 were to maintain your project.
    a. Shiven (shivenk2):
        i. One challenge faced was displaying lots of results at once. Initially we just displayed each row from the advanced queries as plaintext. However, this is visually unappealing and also quite difficult to read. It also makes it hard to copy-paste the data elsewhere, or quickly compare rows, since the columns didn't align. So instead, we spent time building a system to automatically convert from the returned tuples from the SQL query to a HTML table, which would then be formatted with Bootstrap to look much nicer. This was then rendered in a Jinja template to take full advantage of Flask. Some advice from this would be to design your full front-end flow before you create any new functionality, as this took a lot more time than if we had done some things differently from the very start.
            1. Example old format:
            2. Name: Shiven Kumar NetID: shivenk2
            3. Name: Wonjun NetID: wpark26
    b. Wonjun (wpark26):
        i. We had some difficulty initially using the same exact schema between the different tasks as we had made assumptions that slightly differed from person to person on the naming conventions and schema layout. Making a strict consistent schema should be one of the first things a group designing a database should do, even before splitting tasks.
    c. Ravi (ravinp2):
        1. We had some difficulty properly conducting the indexing analysis. Initially we were unaware of the fact that we had to reinitiate the gcp instance every time we had to check and compare the query performance with a new index because the database optimizer was running its own cache mechanism to reduce the cost. After some amount of research from online sources textbooks on indexing analysis, we found out about the issue. So, it is important for a group to first conduct a thorough research and develop a strong understanding of what database optimizations are performed and do the analysis accordingly.
    d. Manav (manavs2):

i. We had some difficulty on the front-end side of the project. Initially, we tried to incorporate modal (web page element that displays in front of and deactivates all other page content) into our front-end. However, we ran into issues with incorporating this into our final project, and as a result, we decided to go with a more simpler user-interface that would still allow the user to interact with the back-end. Thus, I would advise future teams to begin working on the front-end more in advance and getting help from the TAs at office hours.

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

   We planned to add Google Maps interface to our app as the creative feature to make finding restaurant locations easier without a need for another website, but we were unsuccessful in implementing the feature due to lack of time. Furthermore, we planned on having a feature that allowed the user to search for restaurants that were open at a particular time; however, we were unable to implement this feature due to lack of time.

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

   We can add the Google Maps feature to make it easier for users to find restaurants listed. Also, we can add more methods of filtering and viewing the different tables, because right now only a small subset of operations and fields are supported. We should also add the functionality to filter by opening/closing times (for restaurants with that information available) since that was one of the major initial goals. Finally, certain ID fields can be auto-incremented to maintain its primary key status and this would ideally be reflected in the schema by adding the auto-increment property to these fields.

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

    Division:
    Wonjun
    - GCP Hosting
    - Database schema management

- Data generation
- Data conflict resolution

Ravi

- Database indexing analysis
- Initial front-end design and development of midterm demo (collaborated with Manav)
- Creation/implementation of advanced queries + triggers + stored procedures (collaborated with Manav)
- Advanced database program integration to the gcp instance

Manav

- Initial front-end design and development of midterm demo (collaborated with Ravi)
- Creation/implementation of advanced queries + triggers + stored procedures (collaborated with Ravi)
- Video creation

Shiven

- Data fetching script for restaurants (Google Maps API)
- Barebones framework for webapp
- Front-end redesign and new functionalities for final demo (Bootstrap)
- GitHub releases

In advance to our deadlines, we would set up a Discord call where we would distribute the work amongst each other evenly. Furthermore, we would have weekly check-up calls with each other to see everyone's progress and see if anyone needed any help with their part. This setup allowed us all to work productively and efficiently to meet the demands of the project as well as still get help when we needed it. In addition, before major project milestones we would all hop on a call together and group code to maximize productivity and minimize blocking issues, as well as use all of our creativity combined.