Project 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 planned to make a recommendation system to find the best route, flight, and airline for a given destination. However, when we created our advanced query and store procedure, our recommendation system recommends airlines only, but we also created a query to check the delay situation based on state instead of a specific location. Moreover, we mentioned in our proposal that we will include some route visualizations in our project for the creative part. However, we did not include the creative part in our project.

Discuss what you think your application achieved or failed to achieve regarding its usefulness.
We mentioned that we will make a recommendation system in usefulness. We actually made a simple recommendation system to recommend which airline is better. Even though it was not what we expected(a complex route recommendation system), we got it done. However, we fail to visualize paths between different cities on a map. Besides these, we enabled CRUD(create, read, update, delete) for users which we did not mention in our proposal.

Discuss if you change the schema or source of the data for your application.
We did not change the source of data and schema for our project when we finished stage two, but we made some changes when we applied our schema. We will mention those changes in the next part.

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 modify our ER diagram when we finished it, but we made some changes when we applied our table. The first change is that we did not create the CancelledFlight table because we want to focus on flight delay instead of cancellation. This change made our database less complicated. The second change is that we picked more variables as primary keys when we applied FlightSchedule, FlightActual, and DelayedFlight tables. In our schema, we planned to use FlightNumber as the primary key for these three tables. However, FlightNumber is also a foreign key for some tables, so we cannot use it as the single primary key. After these changes, we can have entries which are absent in FlightSchedule in the FlightActual or DelayedFlight table. Consequently, we think our final application is better than our original design.

Discuss what functionalities you added or removed. Why?
We enabled CRUD, create, read, update, delete, and search for users because these functions are required by the project. We also enable two tabs for users to view which airline is better and

which state has higher average delay time. We made these because we want to provide some recommendations for our users. However, we removed visualizations in our project because we do not have enough time to finish it.

Explain how you think your advanced database programs complement your application.
We used two advanced queries to find the average delay time in each state and total delay time for each airline. Then we applied two stored procedures to label each airline and state, such as good choice and bad choice for different airlines. We think these applications are useful because they can provide more clear and easy to understand results for users instead of a lot of numbers. Also, we can use the results from these applications to create recommendation systems for users.

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.
When we did the index analysis, we found that it is almost impossible to improve our application by adding a new index. The reason is that we only have a limited amount of non-primary key variables available for index analysis, so improving performance is almost technically impossible for our project. Also, it is really easy to make technical mistakes when applying index analysis. Once we forgot to drop our index, we got the same result when we indexed on another variable. Another technical challenge is about our trigger. Although our trigger technically worked well when we tested it, it is easily replaced by foreign key constraints. As a result, we are afraid that our trigger will not work as we expect for some specific cases. For example, if the input airline does not exist, our trigger may not work. The last challenge is that we do not know how to include two advanced queries in a single store procedure, so we applied two procedures. We create a new table in our procedure. After we finished the query, we found out that we can write another advanced query after creating the table. We can use the advanced query to return some specific results from our new table.

Are there other things that changed comparing the final application with the original proposal?
We do not have other changes.

Describe future work that you think, other than the interface, that the application can improve on.
We think we can redesign our tables because our queries have large cost and take a long running time, but when we did the index analysis, we found that it is almost impossible to improve cost by adding another index. Moreover, we could revise our store procedure. Since we do not know how to put two queries in one stored procedure, we created two procedures. In addition, our project mainly focuses on flight delay for each state and airline. In the future, we could include

more factors in our application such as routes and more specific places. Also, we could add the cancelledFlight table in the future to make our application more useful.

Describe the final division of labor and how well you managed teamwork.
Kelly - Database Design, ER Diagram, Relation Schema, Index Analysis
Tony - Front-end application, Database Implementation
Brandon - Store Procedure, Trigger, Database Implementation