Project Report

1. **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).**

    a. Our basic idea and direction for the project remained the same where we wanted to inform users about the delay probability of the airlines so that they can choose the airlines they want to fly with based on this information. We just added a few functionalities to improve user experience and provide better information about the flight performance ratings which also include the delay probability.

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

    a. Our application allows the user to submit the reviews of the airline which, depending on their credibility, affects the airline ratings. This gives useful information to other users before they decide to book their flights.
    b. Our application incorporated the delay into the flight performance ratings which gets updated every time the flight schedule is changed.
    c. Due to paucity of time and our limited knowledge of frontend we were unable to use the user's location to list the busiest airports near them.
    d. Also, we could not add filters based on the user's preference for flight time, airline, airline ratings etc.

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

    a. We used 2015 Flight Delays and Cancellations | Kaggle as the main data source. This data has been provided by the course staff.
    b. We also scraped the airline review website Airline Quality to fetch real world user reviews for the airlines as seed data.
    c. To the airline table, we added an Overall Performance column which maintains the performance status of the airline over time based on their delays performance and user reviews.
    d. We also added a column to record the credibility of the user to the user review data which we scraped from the internet. Users with high credibility get greater review weightage when we update the flight reviews and the

airline performance. This gives incentive to the user to give reviews more frequently, allowing us to get more accurate airline performance ratings.

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?**

    a. We did not change much in our original ER design, except that we made the Flights table a weak entity of Airline. We realized that there cannot be any rogue flights (mostly) that are not attached to any airline. Hence, we feel this design simulates the real world situation better.

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

    a. We added a review page because it makes more sense to have a way to update reviews based on the current performance. We are only updating the airlines based on past reviews, we want the ability to create new reviews so that the airline performance gets updated whenever a new review is created. This helps us to keep a better track of the airlines as time passes.

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

    a. The trigger helps us to update the flight performance based on the delay probability of the flights. This helps us incorporate the delays also in flight performance which is not usually done in the usual flight reviews pages.
    b. The stored procedure helps us to track and report the performance status of the airline over time. It also helps us to update the flight performance ratings based on the credibility and authority of the users. Reviews from authorized users with high credibility get more weight while calculating the final review ratings of the airlines. This encourages the users to submit more reviews and helps us provide a truer representation of flight performance.

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 where to maintain your project.**

   a. Debugging / Shubhi - One of the most complicated problems we stumbled upon was when we added triggers and stored procedures in our query in Stage 5. While the query that we were running was to update the ‘Flights’ table, we got the error message that the table ‘FlightReview’ does not exist. For a long time we could not make sense of this error as we were not even touching the ‘FlightReview’ table in our query. Only after an hour of thinking and going through the code line by line were we able to identify that the error was in the trigger which was getting called on update for ‘Flights’ table. Thus, we realized that when an error arises which points to a different table than the one our immediate query is working on, then we should check the trigger or stored procedure code which is getting called on performing that query. Eventually, we spotted the problem in our trigger and replaced the error-causing code.
   Also, if you add new columns in the database while building the application, make sure that you have updated the queries in your application to account for the new columns.
   b. Stored procedure / Zehui(Ray) - So in our stored procedure, we were changing the reviews and airline performance based on certain criteria from the users’ reviews. A technical challenge arose when we needed two cursors for our stored procedure. We needed one cursor to find an index of the top airlines, and the other cursor to change information in the ‘FlightReviews’ table. This was confusing because in class we had never created two cursors in one stored procedure. For future students, I would say that the best way to create two cursors in one stored procedure is to use two BEGIN and END clauses, one for each cursor. This will then allow for one cursor to run first, and then finish before the second cursor starts.
   c. GCP connection / Peter - Each team member needs to connect to the project on GCP via IP addresses, which requires a basic knowledge of communication network, and one issue we face is if you first build the connection using the public IP given by the UIUC network, then when you go back home you will not be able to connect since your public IP changes. So you need to add a new public IP to GCP SQL each time you connect to a different network if you want to connect to the database.
   Also, for members other than the project’s creator, they need to get familiar with the MySQLWorkbench to interact with the database.

d. Front end / Zaid - None of the team members were comfortable with frontend coding and had limited experience with it. This led to a steep learning curve in the beginning where we constantly needed to look up what syntax to use for HTML, EJS, and CSS. Also, it was difficult to figure out how the front end HTML connected to the backend javascript code. I consider this a technical challenge because of my unfamiliarity with the front end languages and the emergence of new libraries in NodeJS to improve developer productivity. If another future team did a similar project, I would say to start watching basic front end videos or going through W3School to just get a simple understanding of front end programming before you really start implementing the project. Also, fully understand the libraries provided by the framework that you use for your backend. Everyday new libraries are getting added to the backend frameworks which make it increasingly easy for developers to establish the flow between frontend and backend.

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

    a. Most of what we have is aligned with the stage 1 proposal submission. What we did was added a page where a user can submit a review, this made it possible to change the overall performance of an airline based on what users believed the experience was like.

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

    a. Use ML models like regression to get more accurate delay probabilities by adding more data to the database.
    b. We can also add real time tracking of flights and use machine learning to predict real time delays based on the airline, weather, airport traffic etc. For this we will need to add a lot of data to our system related to the weather, airport traffic, holidays etc.
    c. Create monthly or bi-monthly reports on the airline performance and send it to the airlines as feedback so that they can work on the services which were rated the lowest.
    d. Add billing information so the user can not only view flights but also book them.

e. Also add the condition to update the flights if the airline merges with another airline. Right now we only handle the case where the airline goes bankrupt.

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

a. We all worked together for the ER/UML and project proposal.
b. For the code, we split it up into different parts:
    i. Shubhi worked on the stored procedure as well as putting the database into GCP.
    ii. Zaid and Ray worked on the advanced queries for the delay probabilities as well as the airline performance.
    iii. Peter worked on the trigger, as well as helping with any other tasks we needed to complete.
c. We all worked on the front end together. We managed our teamwork well as we completed different tasks and discussed with each other about what needed to be done before the deadlines.