

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. **We did not include an interactive map or a function that gives you the best route given a starting and ending destination. We also did not include a route-finder that is based on the user's desired route start time/end time. Instead, we just have basic CRUD functions which allow one to look-up certain routes based on route_ids. We also have a function that sorts routes based on the price of the fares. In addition, we have a function that deletes a trip if the route containing that trip is deleted which ensures that all the data we have is correct/up to date.**
2. Discuss what you think your application achieved or failed to achieve regarding its usefulness.
 - a. **Showing the best route based on starting and ending point would have been a useful element, but we did not have it in our application. This was our original goal that we set out to achieve but we found it to be too difficult and complex to achieve as it would require multiple searches across the database and we often got stuck in searching loops with our queries.**
 - b. **However, our application lets the user see all routes on any search condition and sorts by cost which is useful for people attempting to plan a trip. This is done via the advanced query which lets the user sort the routes based on how expensive the fares would be. In addition, the trigger we have implemented allows for the information in our database to be correct and updated every time a certain element is deleted. By removing routes that have no connected components (and are thus impossible to reach). However, this is potentially bad as if new routes are added, it may be valuable to keep those routes in the database as they would become "live" routes again.**
3. Discuss if you changed the schema or source of the data for your application
 - a. **We did not change the schema of the source of our data. Our data came from the TA-recommended list from Kaggle. However, we had to slightly**

modify our data from the method of storage in order to comfortably receive it from our backend (converting integers to strings) and clean the data.

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. **The one major difference between the original design and our design is that we got rid of the relationship between trips and fare and added a many to one relationship between routes and fare. The reason for this was not because the design was wrong but rather that we realized we wanted to do more stuff with routes than trips. Due to this reason we ended up never using the relationship between trips and fare. We also did not realize at the start that a fare is much more associated with a route than it is a trip. Logically, the reason a trip has a fare is because of its respective route. Though our design still would have worked in its original state, it arguably made less sense because of this strange relationship. It was for all of these reasons that we feel the current design with having a relationship between routes and fare instead of trips and fare makes much more sense.**

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

- a. **One functionality that we added was the ability to analyze what routes would be best to delete by providing information on the number of trips that contain the route, and if the route had an expensive fare. We thought adding this feature would make a lot of sense as it would provide the user to make a better educated guess for when they delete routes. One functionality that we removed was the ability to find the shortest path given your location in Sao Paulo. This would have been very cool, but it was just a bit too ambitious for us to pull off as we were already very busy with building the basic functionalities of the website due to the fact that none of us had any web dev experience.**

6. Explain how you think your advanced database programs complement your application.
 - a. **Our advanced programs let the users sort through routes based on how expensive the fare for that route would be and the amount of trips that use that route. This is a useful feature to have as it will allow the user to better understand which routes might be the best to delete as if there is a route that is both cheap and is not used by many trips, this might indicate that the route is bad and should be replaced by something else.**
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. **Aadya: I had to manually clean up data before we could upload it and use it because the data was in Portuguese which involved special characters that SQL/GCP did not recognize. It was a tedious task but it showed the reality of what working with big data is like. Oftentimes, the data is messy/unstructured and we need to prep the data before we can even use it.**
 - b. **Natalie: I had to figure out how to write functions that called all of the backend database endpoints. It was challenging to figure out the format of the data received from the endpoint, and convert that json to an array structure that the HTML code was able to understand and convert into a table. It was only after calling the endpoint and debugging the response through viewing the console output that I was able to convert the data structure correctly.**
 - c. **Matt: I had to debug our procedure which was quite hard because the bugs were quite hard to diagnose. In particular, a lot of our natural joins were not working properly. They would return empty sets in spite of the fact that there were columns with the same names that had the same values. I became even more confused when I would select both the columns that the natural join was on in both tables and noticed that**

they were the complete same. The problem turned out to be that there were special characters that were not visible in the mysql terminal that we were not dealing with when we preprocessed the data. As a result we had to preprocess the data further, and re-add each respective table.

- d. Arul: I had to figure out how to connect the backend of the application to GCP and it was a particular challenge getting the application to connect to the database since I had never worked with JavaScript before. This was also a challenge because sending and receiving data from the database was done primarily via strings and I had to figure out how to decode the input through comma separation and encode the output into something that could be outputted into our tables.**
- 8. Are there other things that changed comparing the final application with the original proposal?**
 - a. Our final application is definitely a lot more simple and straightforward than our original plan. Specifically we had to remove a lot of the originally planned functionality due to time constraints and difficulty in learning how to interact with GCP and create an application. We removed our search features and the intended map of all routes existing. However, we implemented a more in depth and robust search functionality which allowed users to search for routes catered to their needs by cost. This was the most major shift in our application.**
- 9. Describe future work that you think, other than the interface, that the application can improve on**
 - a. We could implement the original features that we mentioned in the project proposal; specifically, we could have implemented a search feature for any user inputted start and end state. Also, we have data on the longitude and latitude of different stops. We could use this data to create a feature that shows the route on a map whenever the user searches it up. We can also clean up the data itself since a lot of it is unformatted and raw.**

- b. One other potential feature is to create a trigger that would delete all routes that have the same start and end point but with a higher cost as if we are deleting a lower cost route, we may want to delete the higher cost routes as well. This depends on whether we intend to function as a search engine or a live tracker of possible routes.**

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

- a. Aadya: upload data to GCP, create tables using DDL commands, manually clean up data**
- b. Natalie: front-end: wrote the React framework and functions/hooks to call the backend endpoints**
- c. Matt: The trigger, procedure, and aided with writing and debugging some of the frontend.**
- d. Arul: Connected to the database on GCP using IP connection. Created backend interaction with GCP via post requests, and created a backend encoding and decoding system of data to pass to GCP.**