Developing this project was a challenging and exciting experience for me. I gained a lot of insights and knowledge working on this project. The project revolved around computing the optimal path a flight has taken from its start destination to the end destination using the dataset from https://www.openflights.org. To kick off, I started by outlining the necessary steps to meet the project requirements and specifications. The next step was to develop an efficient searching algorithm that would be able to optimally compute the valid route between two cities. After researching a couple of searching algorithms, I finally settled on using Breadth-First Search since it suited the problem criteria the most and was relatively more optimal compared to other algorithms such as A*, Depth-First Search, etc.

Breadth-First Search (BFS) is an algorithm for searching a tree data structure for a node that satisfies a given property. It starts at the tree root and explores all nodes at the present depth prior to moving on to the nodes at the next depth level. I used the Java Buffer Reader module to read the contents of the files containing the start and end destination of a flight, where I computed the valid routes a flight would normally take to reach its destination as well as the time taken. I read the input file and the airport file and then split it into two parts, the start city, and the destination city. I then split the airport file into 5 parts, the airport id, the airport name, the airport city, the airport country, and the IATA. Then check to see if the airport city is equal to the start city and if it is, then set the start airport to the airport id, the airport name, the airport city, the airport country, and the IATA. Then I checked to see if the airport city is equal to the destination city and if it is, then it is equally set to the destination airport to the airport id, the airport name, the airport city, the airport country, and the IATA.