Master, Prince Sefa Yeboah
Mateen Adams

TEAM REFLECTION

First and foremost, the problem was to find a valid or optimal route between the two cities. The optimality was going to be judged either in terms of the flights or in terms of the distance covered by the route.

Initially, it was a bit difficult to comprehend everything I had to do in the assignment. So, I had to research and ask a couple of friends who had a better understanding to approach the problem. A site like Stack Overflow was also helpful in coming to a decision on what approach to take in solving the problem. After many deliberations, the Breadth-First Search algorithm was going to be used to solve the problem. The breadth-first search or BFS algorithm is used to search a tree or graph data structure for a node that meets a set of criteria. It begins at the root of the tree or graph and investigates all nodes at the current depth level before moving on to nodes at the next depth level. You can solve many problems in graph theory via the breadth-first search. Data structures such as an Array List and HashMap were going to be used to solve the problem too. The Breadth First Search algorithm was basically going to be used to search a tree data structure for a node that satisfies a given property, that is, the optimal path.

In the progression of the project, the BFS algorithm proved helpful since we were able to compute the valid route. The use of functions came in handy to make the code implementation modular and well-structured as well as the use of good code commenting. An example is the creation of a public class named **routeTaken**() that is used to find the path between the start destination and the end destination. However, the real problem began when there were challenges to compute the optimal path using the Haversine formula. The real reason was we did not understand the formula entirely. So, we had to reach out to a colleague who had knowledge of the use of the formula. Fast forward, a colleague named Iddriss Raaj helped in understanding how the implementation of the Haversine formula was going to work

in the code. After several attempts to compute the distance between two points with given latitude and longitude coordinates, we were able to compute the optimal route/distance.

Despite the various problems and challenges faced trying to find the optimal path, it was a success when we were finally able to compute the optimal path

In conclusion, we have learned as a team that discipline and understanding is crucial in pair programming.