REFLECTIVE REPORT C++ PROJECT

Intermediate Computer Programming

Abubakari Sadik Osman

42112024

November 29, 2022

In this project, we first analyze and break down the problem into segments for easy implementation. We implemented the Airport, Airline, and Route classes with getters and setters for effective access and mutation.

The dataset from the route.csv files contained some filthy data, so we filtered them out for efficient implementation and manipulation.

Next, we implemented ListOfAirlines, ListOfAirports, and ListOfRoutes which return the list of airlines, airports, and routes from their CSV files in an unordered\_map data structure for the ListOfAirlines, ListOfAirports, and ListOfRoutes for easy accessing.

We represented the problem as a Node class that has a start i.e the initial city and country and parent which is initially set to NULL. We use the city and country specified to create an airport which is then used to create the Node class. The Node class method getSuccessors() takes the current or source airport and returns the successor airports. Each of these successors is used in the breadth-first search method explained below.

Finally, To solve the search problem, we implemented a breadth-first search that uses a vector with the push\_back() and front() methods (First-In-First-Out) for its implementation as the airport is being processed or explored to get the successor airports (reachable or destination) airports and add to the queue and remove all the already explored airports from the queue. This is achieved by using input city and country to create an airport instance and comparing the airport ID to the route IDs to get the source airport ID. The source airport ID is then used to get the destination airport IDs and add them to the vector by the push\_back() method. Each of these is compared to the destination airport created from the destination city and country names.

We created methods that have functionalities of writing and reading when found. This allows the user to input cities and countries from the terminal and this data is written to a file called input.csv. The FindRoute class is where most of the functionality of the program dominates.