Gavin O'Neill – Student No: D15127205

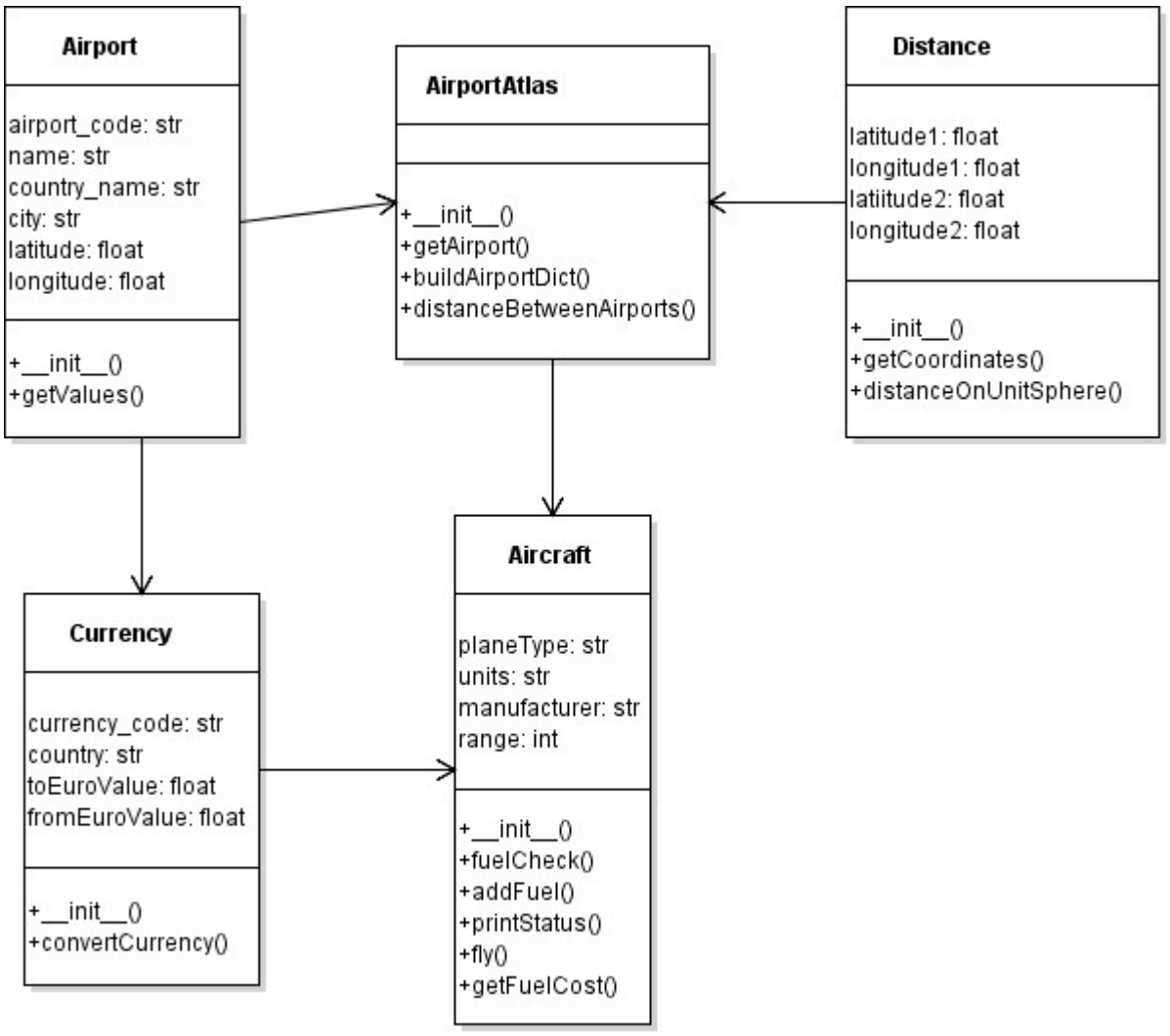
Object Oriented Software Development Project

(1) Function Specification

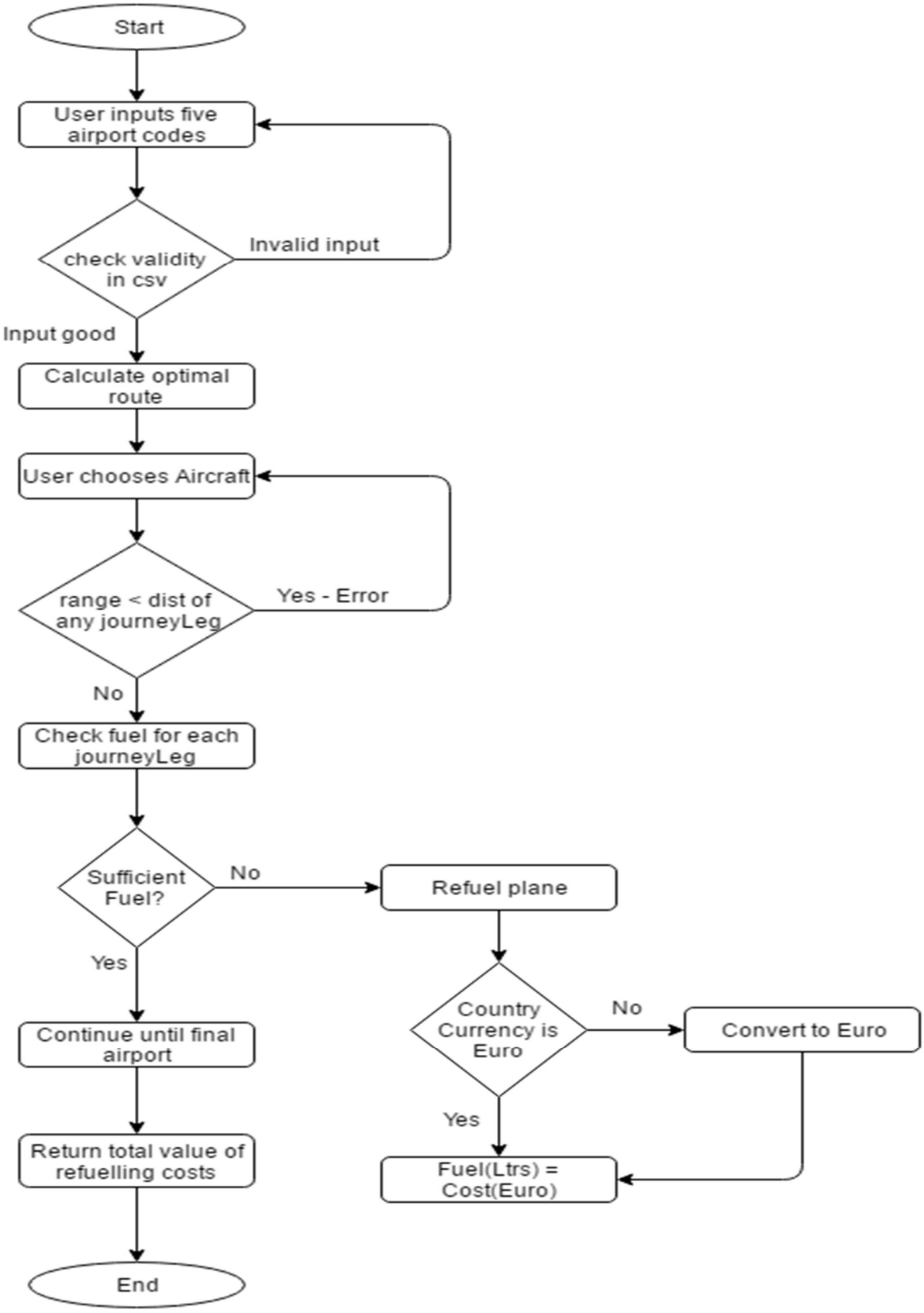
The aim of this project was to design a fuel management application that would take a list of airports from a user, calculate the optimal route for an aircraft to travel to every airport and return to its home base, calculate any/all refuelling costs and return the cheapest journey to the user. Unfortunately, my programming skills are not up to this task at the moment so I have not been able to produce a functioning program. Below is a description of the design that I had hoped to implement if I had been able to.

(2) Design: Assumption, Inputs and Outputs

Class Diagram



Main Algorithm Diagram



My class diagram is designed to display all of the data that will be manipulated and stored by this program. Each class represents a different object type in the system, they each have their own attributes and methods but they are all dependent on one another to function correctly. For example, the AirportAtlas call is what provides the distances between all airports and to do so it requires information from the Airport class such as geographical coordinates which are then passed to the Distance class in order for it to calculate the distances between airports. These distances are then needed by the Aircraft class to ensure that the aircraft selected has the range to fly the chosen route and the distances are also used to check if refuelling is needed, hence allowing us to work out the cost of the journey. Finally, this dependence almost goes full circle as the Currency class also gets country information from the Airport class to see if the values need to be converted or not.

In my main algorithm diagram, I have tried to depict how I see the program working from start to finish including any decisions that need to be made along the way. The five airport codes are entered by the user and a lookup is performed on the csv file, if the data is found then a dictionary is created in the AirportAtlas object and the optimal route can be worked out. Once this is completed, the user can select an aircraft and so long as it's suitable for the journey the system will begin to go through the fuel checks at each airport of the route, refuelling when necessary, adding any refuelling costs together and return the total cost of the journey to the user on completion.

(3) Testing

As my program as a whole is unfinished, the only testing I have are single test files to check if the code will produce the output I expect. The test files have the same name as the file they are testing with the prefix 'test' added to the filename. What I have coded so far runs without errors but sometimes doesn't produce the output I expect or else I do not have the input I need from the csv to run a proper test.