Individual Design Assignment

Name: Bigyan Adhikari (CS Student)

UTA ID: 1001398922

CSE-3310-002

**Design Report**

1. **Problem Statement**  
     
   Locating places on earth has been a really good feature of present world that enables airplanes, vehicles, people and other technological devices to navigate.   
   We want to build a program that is capable of finding all airports within a given radius of a given airport.   
   In order to locate close airports within given radius, we will need to have the distance between the given airport and other airports, but we might not have distance information of a particular airport with other airports all the time. On the other hand, we can fix this issue by evaluate distance between two places on earth by using their geological values.   
   Every place on earth will have its longitude and latitude. If we are able to use this geological value, we can calculate the distance between any points by using math equation and list the airports that are close to a given airport.
2. **Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| Individual Assignment Phase | Est Time | Time Taken | Reason it took long or not long than the estimated time. Description |
|
| Thinking about the approach | 5.0 | 5.0 | On time |
| Writing distance function | 120.0 | 60.0 | Professor provided a reference link which has math equations for finding distance |
| Opening file and tokenizing elements | 60.0 | 300.0 | Thought easier but figured many constraints. |
| Had to tokenize elements either by " " " or " , " comma in different cases |
| Using distance function and printing statements | 30.0 | 60.0 | stold conversion error occurred. took longer than expected |
| Fixing and Handling corrupt/misplaced tokens | 60.0 | 10.0 | I thought it would take longer but after using exception throw, it was easier to handle this error |
| Comments | 30.0 | 30.0 | On time |
| Make-file | 10.0 | 10.0 | On time |
| Total Time Taken in Minutes | 275.0 | 435.0 | 160 mins longer than expected |
| Total Time Taken in Hours | 4.6 | 7.3 | 2.67 hrs. longer than estimated |

1. **Requirement Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| **REQ ID** | **F/NF** | **DESCRIPTION** | **CLASS/METHOD** |
| 100 | F | Program must use C or C++ standard library | int main() |
| 101 | F | Program must use the values from given data file airports.dat | int main() |
| 102 | F | Program shall take user input as four ICAO letters and radius from command line tool | int main() |
| 103 | F | Program will execute until user enters negative radius | int main() |
| 104 | F | Program shall tokenize the lines in data file in order to acquire corresponding values | stringstream limitLine(fileLine); |
| 105 | F | All the tokenized strings should be stored for future reference | getline(limitLine, name, '"'); |
| 106 | F | Some tokenized strings need to be converted into integers or double value | stold(strings); |
| 107 | F | Error tokens should be handled if they occur | int main() |
| 108 | F | Program will have functions to convert geological coordinates into distance | long double distance(X4 long double) |
| 109 | F | Program can have some string tools like erasing begin and end characters in order to remove some characters | string.erase(): |
| 110 | F | Program needs two loops to traverse entire lines | while (! EOF ) |
| 111 | F | One while loop will be used to catch given airport position | int main() |
| 112 | F | Another loop will be used to compare distance between given airport and other airports | int main() |
| 113 | F | Program shall return all airports that lie in a given radius | int main(), cout<< |
| 114 | F | Program shall have case handling if it cannot convert strings into integers | try{this} catch(exception} |
| 115 | F | Program shall close all open file after using it | int main(), file.close(); |
| 116 | NF | Program shall be delivered in a tarball with all source codes | - |
| 117 |  | Program will inform users how many conversions error occurred | - |
| 118 | NF | Program must be readable | - |
| 119 | NF | Program code shall have comments | - |

1. **Design**

**A picture containing table

Description automatically generated**

**Class Diagram**

|  |
| --- |
| + id,  + name,  + city,  + country,  + iata,  + icao, + Latitude,  + Longtitude,  + altitude,  + time,  + dst,  + tz,  + type,  + source;  + namec, + cityc, + countryc,  + iatac, + icaoc;  + userAirportLat  + userAirportLong  + radiusArea + radius  + userICAO |
| + long double convertToRadians(degree: long double)  + long double distance(userAirportLat: long double, userAirportLong: long double, dataAirportlat: long double, dataAirportlong:long double) |

1. **Alternatives**

|  |  |
| --- | --- |
| Approach 1 | Approach 2 |
| Evaluating distance using 2D vector  Design a 2D vector that stores all the vectors that has airport data  Storing the given airport values in some data structure.  Then we can traverse through the 2D vector, compare distance between the airports and print all the airport that lies in the given radius. | By using complete OOP paradigm.  Would lead to create multiple objects.  Multiple objects are hard to abstract  And would take more time to fix error if we used OOP.  Would take me too much time to think about OOP principles rather than solving deep problems of the progeam. |

I realized that all the lines couldn’t be tokenized in a similar way with other lines.

This might create error strings and puts the values in vector in an incorrect order.

While accessing that vector, it can cause error while performing string to integer conversions.

So, I used two loops to solve this without making 2D vector or objects. Instead, I used two loop to solve this.

1. Constraints
2. All the lines in the data file doesn’t separate values in fixed order

Case 1: Lines 4295 and 4296 in airport.dat

Line 4295 has 14 “,“ comma

Line 4296 has 13 “,” comma

If we use “ , “ as key to generate tokens, it will place tokens in different order.

A picture containing man, holding

Description automatically generated

I found this constraint by calculating total number of “,” and dividing it with total number of lines. When I divide the total number of “,” lines by total lines, the result was a decimal integer.   
Problem: If I used “ , “ to separate values I would miss full name of the airport and if I used quotations ‘ “ ‘ I wouldn’t be able to get values for latitude and longitude.

My approach to solve this:

Firstly, I tokenized half of the lines using quotations ‘ “ ‘ and rest of the lines by comma “ , “ . I tried to insert maximum tokens into correct values but everything in the universe is not perfect and easy. This constraint would affect converting latitude and longitude and during integer conversion, there would have been problem. I implemented exception throw clause in this program to escape invalid string to integer conversion error as below:

A screenshot of a cell phone

Description automatically generated

Fig: How values are tokenized. Safely catching full name of the airport.

A screenshot of a cell phone

Description automatically generated

Fig. How exception throw is implemented   
If error strings arrive for latitude and longitude tokens, escape the stold execution.

1. Null values \N

A screenshot of text

Description automatically generated

This won’t affect the output because above tokenizing procedure and exception throw will handle this.

1. Some airports name have “ , “ comma. If comma is used to tokenize line, there’s chance of missing airports full name  
   A screenshot of a cell phone

   Description automatically generated

The tokenizing procedure shown in source code above will solve this issue.

1. Chance of printing the user given airport   
   Case: For airport KDFW, its printing its own distance.

A screenshot of a cell phone

Description automatically generated

Solved this issue using simple if statement.

A screenshot of a cell phone

Description automatically generated

After solved:

A screenshot of a cell phone

Description automatically generated

1. Source code

Source Code is in tarball

1. Analysis

Test Case 1: Program gives ICAO, full name of Airport and distance from the given airport.

A screenshot of a cell phone

Description automatically generated

Test Case 2: Incorrect latitude or longitude tokens found and were skipped to avoid stold error. 12 possible error avoided.

A screenshot of a cell phone

Description automatically generated