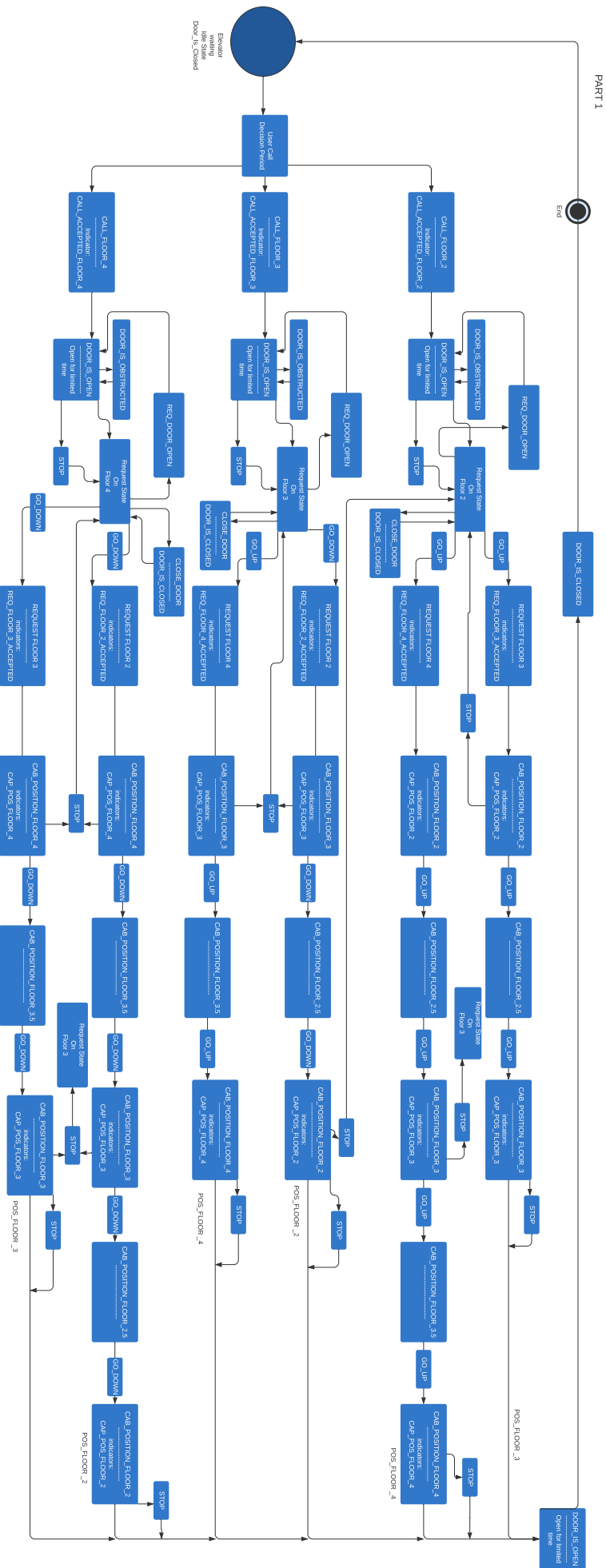


# State Diagram for 1980s KONE roped hydraulic elevator, @ Söderhamn Hospital,



**Descriptions: Based on my assumptions:**

1. Person enters through the door after using elevator control panel outside
2. Person is allowed to use the keys to go up, go down, open door, stop and use bell keys
3. After door is closed and user input is received, the elevator starts to move
4. After elevator achieves or passes the floor level sensor, person can stop at that floor.
5. Person can change decisions after pressing the stop key.
6. Elevator floor position and motion is shown in the indicator lights.

## PART 2

1. The Elevator Controller should control elevator to move up according to user destinations and elevator initial position (GO UP= TRUE &88 REQ. FLOOR -4)is received). (For going to floor -4)
2. The Elevator Controller should control elevator to move down according to user destinations and elevator initial position (GO DOWN= TRUE &88 REQ. FLOOR -2)is received). (For going to floor -2)
3. The Elevator Controller should control elevator to stop and reset the user inputs if (STOP= TRUE)is received through the stop button.
4. The Elevator Controller should control elevator to open the door if (OPEN DOOR= TRUE) is received through the open-door button.
5. The Elevator Controller should also control elevator to open the door if (OPEN DOOR= TRUE &88 CALL\_FLOOR\_>= TRUE)is received through the open door button from outside.
6. The Elevator Controller should control elevator to close the door if (CLOSE DOOR= TRUE) is received through the close door button.
7. The Elevator Controller should control elevator to display arrival position according to its floor position (display 2 if POS. FLOOR= 2) and so on.

Elevator link: [https://www.youtube.com/watch?v=tJ9Qf\\_UhbXY](https://www.youtube.com/watch?v=tJ9Qf_UhbXY)

UTA, SPRING 2020

# Some Requiriments

The Elevator Controller should control elevator to move up according to user destinations and elevator initial position (GO UP = TRUE && REQ\_FLOOR\_4) is received. (For going to floor 4)

The Elevator Controller should control elevator to move down according to user destinations and elevator initial position ( (GO DOWN = TRUE && REQ\_FLOOR\_2) is received. (For going to floor 2)

The Elevator Controller should control elevator to stop and reset the user inputs if (STOP= TRUE) is received through the stop button.

The Elevator Controller should control elevator to open the door if (OPEN DOOR= TRUE) is received through the open door button.

The Elevator Controller should also control elevator to open the door if (OPEN DOOR= TRUE && CALL\_FLOOR\_2= TRUE) is received through the open door button from outside.

The Elevator Controller should control elevator to close the door if (CLOSE DOOR= TRUE) is received through the open door button.

# UTA, SPRING 2020

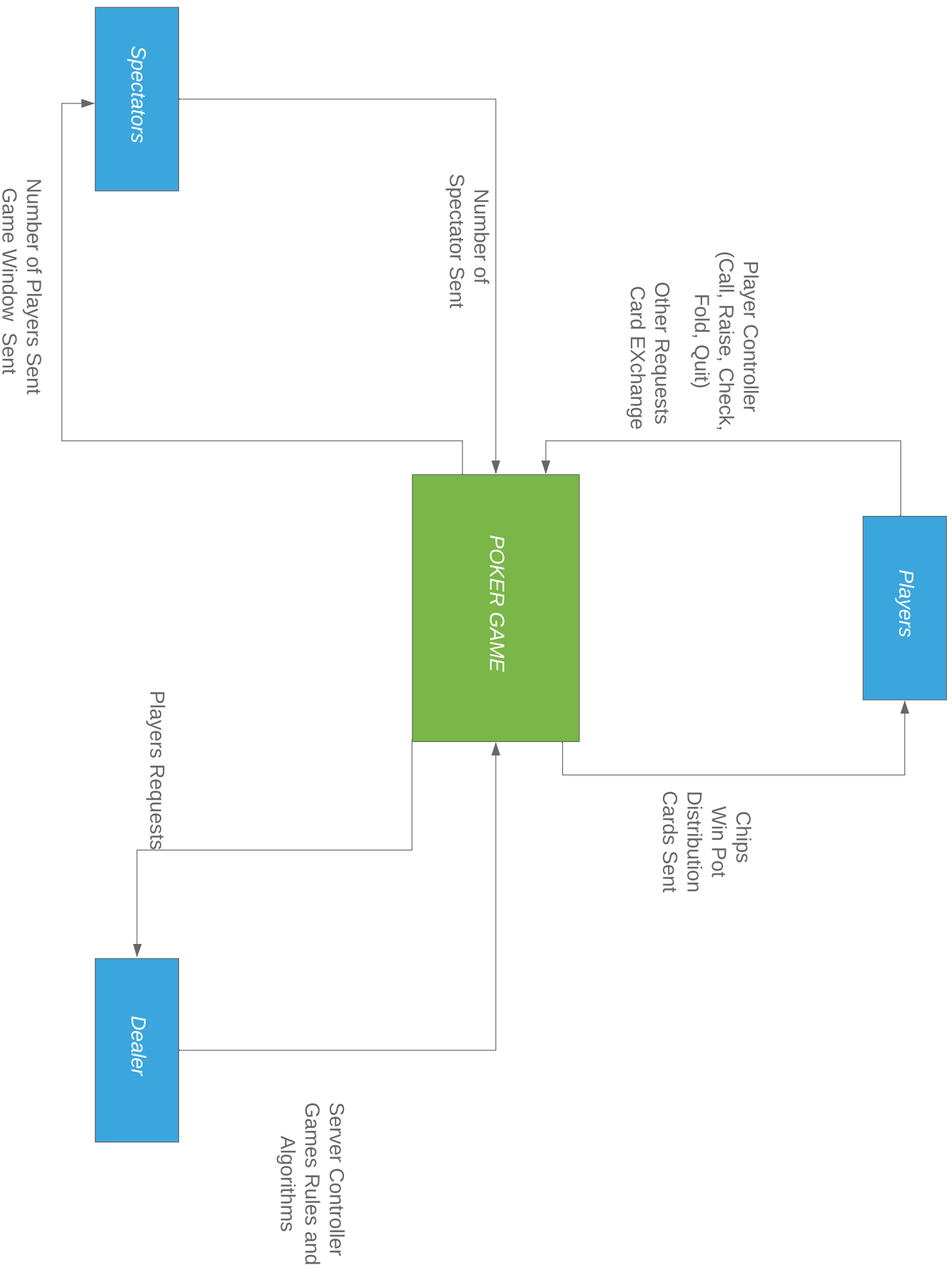


Fig: Context Diagram-Poker-  
Group10

|   |  |
|---|--|
| <p><b>UI Input button</b></p> <p>Raise<br/>Check<br/>Fold<br/>Call<br/>Raise<br/>play button<br/>Instructions button<br/>spectator button<br/>quit button<br/>Player name entry box<br/>Player ready button</p> | <p><b>Object ID in C++</b></p> <p>RaiseBin<br/>CheckBin<br/>FoldBin<br/>CallBin<br/>RaiseBin<br/>playBinClicked<br/>InstructionsBinClicked<br/>SpectatorBinClicked<br/>QuitBinClicked<br/>PlayerNameBox<br/>PlayerReadyBin</p> |
| <p><b>Functions/Algorithms</b></p> <p>hasEqualHand<br/>hasHighestHand</p> <p>has the highest hand<br/>has the highest rank card</p>   | <p>Checks players cards<br/>have equal hand rank<br/>Finds out which player<br/>has the highest rank card</p>  |
| <p><b>State/Process</b></p> <p>State/Process</p>  | <p>State/Process</p>   |
| <p><b>Buttons/Control</b></p> <p>Buttons/Control</p>  | <p>Buttons/Control</p>   |



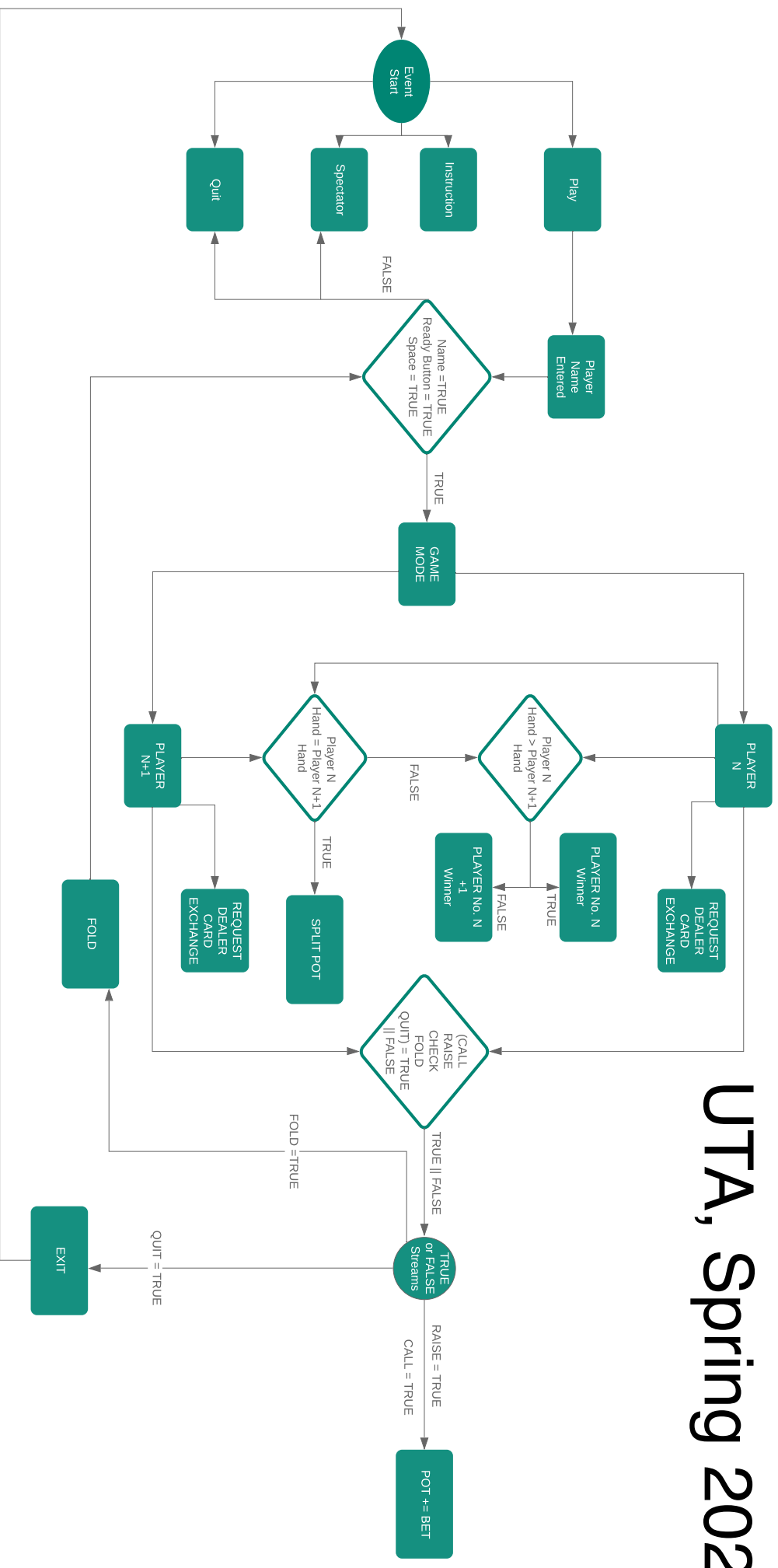


Fig: Event Diagram - Poker Game-  
Group 10

## Individual Design Assignment

Name: Bigyan Adhikari (CS Student)

UTA ID: 1001398922

CSE-3310-002

## Assignment 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.

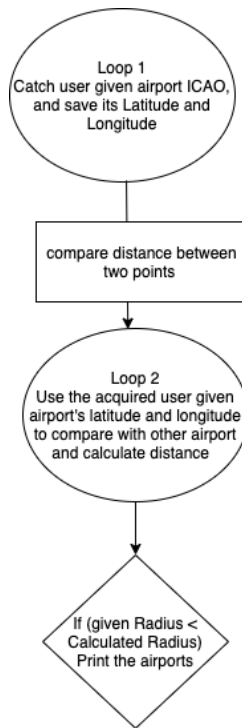
### 2. Schedule

| 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.<br>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  |

### 3. 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  | -                                    |

## 4. Design



## Class Diagram

```
+ id,  
+ name,  
+ city,  
+ country,  
+ iata,  
+ icao,  
+ Latitude,  
+ Longitude,  
+ altitude,  
+ time,  
+ dst,  
+ tz,  
+ type,  
+ source;  
+ namec,  
+ cityc,  
+ countryc,
```



|  |
|--|
| + iatac,<br>+ icaoc;<br>+ userAirportLat<br>+ userAirportLong<br>+ radiusArea<br>+ radius<br>+ userICAO  |
| + long double convertToRadians(degree: long double)<br>+ long double distance(userAirportLat: long double,<br>userAirportLong: long double, dataAirportlat: long double,<br>dataAirportlong:long double) |

5. Alternatives

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

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.

## 6. Constraints

- I. 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.

```
4289 5581,"Namsos Heknesøra Airport","Namsos","Norway","OSY","ENNM",64.472198486328,11.57859992981,7,1,"E","Europe/Oslo","airport","OurAirports"
4290 5582,"Mo i Rana Airport, Røssvoll","Mo i Rana","Norway","MQN","ENRA",66.363899230957,14.301400184631,229,1,"E","Europe/Oslo","airport","OurAirports"
4291 5583,"Rørvik Airport, Røm","Rørvik","Norway","RVK","ENRM",64.838302612305,11.14610004425,14,1,"E","Europe/Oslo","airport","OurAirports"
4292 5584,"Røst Airport","Røst","Norway","RET","ENRS",67.527801513672,12.103300094604,7,1,"E","Europe/Oslo","airport","OurAirports"
4293 5585,"Sandane Airport (Anda)","Sandane","Norway","SDN","ENSD",61.830001831055,6.1058301925659,196,1,"E","Europe/Oslo","airport","OurAirports"
4294 5586,"Sogndal Airport","Sogndal","Norway","SOG","ENSG",61.156101,7.13778,1633,1,"E","Europe/Oslo","airport","OurAirports"
4295 5587,"Svolvær Helle Airport","Svolvær","Norway","SVJ","ENSH",68.243301391602,14.669199943542,27,1,"E","Europe/Oslo","airport","OurAirports"
4296 5588,"Sørkjosen Airport","Sørkjosen","Norway","SOJ","ENSR",69.786796569824,20.959400177002,16,1,"E","Europe/Oslo","airport","OurAirports"
4297 5589,"Vardø Airport, Sværnes","Vardø","Norway","VAV","ENSV",70.355400085449,31.044900894165,42,1,"E","Europe/Oslo","airport","OurAirports"
4298 5590,"Værøy Heliport","Værøy","Norway","VRY","ENVR",67.654555,12.727257,12,1,"E","Europe/Oslo","airport","OurAirports"
4299 5591,"Bydgoszcz Ignacy Jan Paderewski Airport","Bydgoszcz","Poland","BZG","EPBY",53.096801757799994,17.9776992798,235,1,"E","Europe/Warsaw","airport","OurAirports"
4300 5592,"Łódź Władysław Reymont Airport","Łódź","Poland","LCJ","EPLL",51.7219009398099996,19.3908988993,604,1,"E","Europe/Warsaw","airport","OurAirports"
4301 5593,"Åre Östersund Airport","Östersund","Sweden","OSD","ESNZ",63.194400787354,14.50030040741,1233,1,"E","Europe/Stockholm","airport","OurAirports"
4302 5594,"Hagfors Airport","Hagfors","Sweden","HFS","ESOH",60.02009963989258,13.578900337219238,474,1,"E","Europe/Stockholm","airport","OurAirports"
4303 5595,"Karlstad Airport","Karlstad","Sweden","KSD","ESOK",59.4446983336999996,13.337400436400001,352,1,"E","Europe/Stockholm","airport","OurAirports"
4304 5596,"Torsby Airport","Torsby","Sweden","TYF","ESST",60.1576004028,12.991299629199998,393,1,"E","Europe/Stockholm","airport","OurAirports"
4305 5597,"Ängelholm-Helsingborg Airport","Ängelholm","Sweden","AGH","ESTA",56.29610061645508,12.847100257873535,66,1,"E","Europe/Stockholm","airport","OurAirports"
4306 5598,"Storuman Airport","Möhed","Sweden","SOD","ESUD",64.86089935302734,17.60659996032715,915,1,"E","Europe/Stockholm","airport","OurAirports"
```

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:

```
//we will look the entire file data
while (!inFile.eof())
{
    getline(inFile, fileLine); //Tokenizing entire file line
    stringstream limitLine(fileLine); //stringstream operates in string, will help to tokenize strings into streams
    getline(limitLine, id, ''); //Tokenizing airport id
    getline(limitLine, name, ''); //Tokenizing airport name
    getline(limitLine, namec, ''); //Tokenizing comma " , " after airport name
    getline(limitLine, city, ','); //Tokenizing airport city
    getline(limitLine, country, ','); //Tokenizing airport country
    getline(limitLine, iata, ','); //Tokenizing airport IATA
    getline(limitLine, icao, ','); //Tokenizing airport ICAO
    // cout<<icao<<endl;
    // removing quotation " " from acquired ICAO
    //It will transform "KDFW" to KDFW so it will be easier to compare with user input ICAO
    icao.erase(remove(icao.begin(), icao.end(), '\\"'), icao.end());

    getline(limitLine, Latitude, ','); //Tokenizing airport latitude
```

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

```

//It will transform "KDFW" to KDFW so it will be easier to compare with user input ICAO
icaonxt.erase(remove(icaonxt.begin(), icaonxt.end(), '\\'), icaonxt.end());
getline(limitLine, Latitudenxt, ','); //Tokenizing airport latitude

//After we catch ICAO from data file we will store its latitude and longitude
try{
    dataAirportlat = stold(Latitudenxt);
}
catch(exception &err)
{
    continue; // Couldnt execute stold(Latitudenxt)? Caught error? Good. Continue though!
}

getline(limitLine, Longitudenxt, ',');
try{
    dataAirportlong = stold(Longitudenxt);
}
catch(exception &err)
{
    continue; //Couldnt execute stold(Longitudenxt)? Caught error? Good. Continue though!
}

//Findind the distance between two points
long double radiusArea = distance(userAirportLat, userAirportLong, dataAirportlat, dataAirportlong);

```

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

## II. Null values \N

```

313 315,"Weelde Air Base","Weelde","Belgium",\N,"EBWE",51.394798278808594,4.960189
314 316,"Zoersel (Oostmalle) Airfield","Zoersel","Belgium","OBL","EBZR",51.264702,
315 317,"Flugplatz Bautzen","Bautzen","Germany",\N,"EDAB",51.193611,14.519722,568,
316 318,"Altenburg-Nobitz Airport","Altenburg","Germany","AOC","EDAC",50.981945,12
317 319,"Dessau Airfield","Dessau","Germany",\N,"EDAD",51.831694,12.190962,187,1,
318 320,"Eisenhüttenstadt Airfield","Eisenhuettenstadt","Germany",\N,"EDAE",52.195
319 322,"Großenhain Airport","Suhl","Germany",\N,"EDAK",51.30805587768555,13.55555
320 323,"Merseburg Airport","Muehlhausen","Germany",\N,"EDAM",51.3630556,11.940833
321 324,"Halle-Opplin Airport","Halle","Germany",\N,"EDAQ",51.552223,12.053889,348,
322 325,"Riesa-Göhlis Airport","Riesa","Germany","IES","EDAU",51.2936096191,13.356

```

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

- III. Some airports name have “ , ” comma. If comma is used to tokenize line, there's chance of missing airports full name

```
ENEV Harstad/Narvik Airport, Evenes 4665.68
ENRY Moss Airport, Rygge 4856.45
ENSB Svalbard Airport, Longyear 4295.12
ENTC Tromsø Airport, 4664.61
ENTO Sandefjord Airport, Torp 4847.63
WAPP Pattimura Airport, Ambon 8891.53
YMPC RAAF Williams, Point Cook Base 8997.28
KPBF Pine Bluff Regional Airport, Grider Field 306.552
KHUF Terre Haute Regional Airport, Hulman Field 705.588
ENOV Ørsta-Volda Airport, Hovden 4622.92
ENFB Oslo, Fornebu Airport 4832.67
ENRA Mo i Rana Airport, Røssvoll 4693.12
ENRM Rørvik Airport, Ryum 4671.26
ENSS Vardø Airport, Svartnes 4852.57
YLVT RAAF Williams, Laverton Base 8996.5
HSFA Paloich Airport, Heliport 7956.16
```

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

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

```
Enter a ICAO and the Radius:
KDFW 50.0

KDAL Dallas Love Field 11.3301
KDFW Dallas Fort Worth International Airport 0
```

Solved this issue using simple if statement.

```
if(radiusArea<radius){
    if (userICAO.compare(icaonxt)!=0){ //If user input ICAO comes, dont print that airport
        cout<<icaonxt<<" "<<namenxt<<" "<<radiusArea<<endl;
    }
}
inFilenxt.close();
cout<<"\n"<<endl;
```

After solved:

```
Enter a ICAO and the Radius:
KDFW 50.0

KDAL Dallas Love Field 11.3301
KFTW Fort Worth Meacham International Airport 19.5512
KNFW NAS Fort Worth JRB/Carswell Field 25.0121
KADS Addison Airport 12.691
KGGY Arlington Municipal Airport 16.4095
KTKI Collin County Regional At Mc Kinney Airport 32.36
KRBD Dallas Executive Airport 17.8705
KAFW Fort Worth Alliance Airport 17.4367
KHQZ Mesquite Metro Airport 31.2156
KDTO Denton Municipal Airport 22.935

Enter a ICAO and the Radius:
```

## 7. Source code

Source Code is in tarball

## 8. Analysis

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

```
ENRY "Harstad/Narvik Airport, Evenes" 4665.68
ENV "Moss Airport, Rygge" 4856.45
ENSB "Svalbard Airport, Longyear" 4295.12
ENTC "Tromsø Airport," 4664.61
ENTO "Sandefjord Airport, Torp" 4847.63
WAPP "Pattimura Airport, Ambon" 8891.53
YPMC "RAF Williams, Point Cook Base" 8997.28
KPBF "Pine Bluff Regional Airport, Grider Field" 306.552
KHUF "Terre Haute Regional Airport, Hulman Field" 705.588
ENOV "Ørsta-Volda Airport, Hovden" 4622.92
ENFB "Oslo, Fornebu Airport" 4832.67
ENRA "Mo i Rana Airport, Røssvoll" 4693.12
ENRM "Rørvik Airport, Ryum" 4671.26
ENSS "Vardø Airport, Svartnes" 4852.57
LYVT "RAF Williams, Laverton Base" 8996.5
HSFA "Palaoich Airport, Heliport" 7956.16
```

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

```

//After we catch ICAO from data file we will store its latitude and longitude
try{
    dataAirportlat = stold(Latitudenxt);
}
catch(exception &err)
{
    cout<<"Couldnt execute stold(Latitudenxt)? Caught error? Good. Continue though!"<<endl;
    continue; // Couldnt execute stold(Latitudenxt)? Caught error? Good. Continue though!
}

getline(limitLine, Longtitudenxt, ',');
try{
    dataAirportlong = stold(Longtitudenxt);
}
catch(exception &err)
{
    cout<<"Couldnt execute stold(Longtitudenxt)? Caught error? Good. Continue though!"<<endl;
    continue; //Couldnt execute stold(Longtitudenxt)? Caught error? Good. Continue though!
}

```

[illegible]



Poker Hand Determining Code: <https://github.com/bigyan313/pokerHandRank.git>

