Airport Simulation

University of North Texas

CSCE2110

Section 204

Ajay Jayanth

Eduardo Garcia

Gonzalo Franco

Ivan Garcia

Nicolas Hidalgo Rotunno

**Introduction:** The main purpose of the project is going to be a simulation of an airport where there are going to be a couple CSV files to be read and have data stored on them. The following CSV files are going to be structs called Plane which is going to store:

* ID ,
* Maker,
* Model,
* LastMaint,
* LastMaintA.

PlaneType Is going to store

* Maker
* Model
* flyingspeed
* Ground Speed

PLaneSeats is going to store

* Maker
* Model
* SeatType
* NoOfSeats.
  + There is going to be a relationship between maker and model on these 3 structs

FLightLeg is going to store

* FLNO
* Seq
* FromA
* ToA
* DeptTime
* ArrTime
* Plane

Airport is going to store

* Code
* City
* State
  + FromA, ToA, DeptTime are going to related to code from Airport and ID from Plane

FlightLegInstance is going to store

* Seq
* FLNO
* FDATE
* ActDept
* ActArr
* Pilot

Flight is going to store

* FLNO
* Meal
* Smoking

Pilot is going to store

* FLNO
* FDate
  + Seq, FLNO and FDate are going to be relate to FLNO and seq from FlightLeg and FlightInstance
* Reservation is going to store
* PssID
* FLNO
* FDAte
* FomA
* ToA
* SeatClass
* DAteBooked
* DateCanceled

Passenger is going to store

* ID
* Name
* Phone

Text

Description automatically generated

There is going to be the following connection between all of them. The main purpose of the project is going to be to take the key of the files and that way we are going to be able to access to the data and store it into the variables.

Hash Implementation: The type of hashing that we are going to be using is going to be stringing hashing. This is going be by creating a hash function with parameter int. Then, the key is going to be a name. A for loop is going to iterate through each work and it is going to take the highest value of the ASKII table and multiply it by the hash number + the number of iterations and all of that is going to be modular by the bucket which is 25. That is going we are going to access to the key and the values.

**Part 1:** The way that the file is being read it is going to bbe by using 4 different files. The main is going to have parameters argc and argv in order to read the files. There is going to be a for loop starting at the argument 2 so it can read the second argument given. A file called read\_from\_file is going to be created so that we can read the file, and then there is going to be a parameter argc > 1 so that it can get access to the name of the files where the CSV’s are being stored. WE are going to check if the file fails to open or not, and in case it does there is going to be a message saying that the file failed to open. After doing that we are going to called the function files which is where all the files are going to be stored it and read it.

The function files is going to to have a parameter variable called read\_from\_file passed by reference. Them, a 2D vector with parameter string is going to create a variable called data. The way that the files are going to be read it is going to be through a while loop with parameter read\_from\_file and a line that is going to store the lines from the files. This is going to be the main file that is going to be in charge of reading the files and store the variables. A variable called line with argument “,” that is going to be added is going to help us out to read the CSV files. Another vector is going to be created called words which is going to be in charge of storing the words from the csv file. StringStream is going to help us to read and save the variables from the files and the way we are creating it is by calling the function reading\_lines and passing the variables line into the actual strings. Once that is done a final while loop is going to be used to read the lines, the words and the comas from the csv files and then all of that is going to be storesd into words by using push\_back with parameter word which is. The vector declared before. A final for loop is going to be in charge of printing the words from the files.

**Part 2:** The logic that this program is using going to be using is going by using a class of each object. Inside of that class 3 different private data structs are going to be stored in order to save the data types previously declared. This is going to give us more flexibility at the time to manipulate the variables because in order to have variables connected with each other we can just call data objects that are going to store the information of each data type. This functionality is going to help us out to actually avoid inheritance. After doing that 3 types of tables are going to be used with parameter bucked, and that is going. Be the way to access and call the variables. (**PART 2, THE CODE AND THE WIKI FROM PART 2 WAS NOT SUBMIT IT.)**

**Part 3: (PART 3, THE CODE AND THE WIKI FROM PART 2 WAS NOT SUBMIT IT.)**

**Part 4 ajay:**

**PART 5: (PART 5, THE CODE AND THE WIKI FROM PART 2 WAS NOT SUBMIT IT.)**