Team 8: Dāda Bäce
Adam Nash awn10@pitt.edu
Jason Henriquez jth79@pitt.edu
Ethan Dewit erd56@pitt.edu

Team 8 Report

Features & Descriptions

Before Starting

- Before starting the application, a password and username needs to be added to the ‘info.java’ file so that a connection can be created.

Menu Interface

The program starts by asking the user to select Administrator or Customer. Depending on their chosen role, the user is then displayed a list of actions for their respective role. The sections below describe the possible actions that can be chosen by the Customer and Administrator roles.

Customer Interface Features

#1 Add Customer

- User is prompted to supply all the necessary fields for a new customer: *salutation* (Mr/Mrs/Ms), *first name, last name, address (street, city, state), phone number, email address, credit card number, credit card expiration date, frequent miles*
- User will be prompted for each field
- After all fields are supplied, new entry is inserted in the customer table

#2 Show Customer Information

- User is prompted to supply first and last name of existing customer
- If customer does not exist, an error message is shown
- If customer exists, all of that customer’s information is displayed

#3 Find Price For Flights Between Two Cities

- User is prompted to supply two cities
- If cities are valid, and flights exist, high and low prices are printed for city 1 to city 2. High and low prices are also printed for city 2 to city 1

#4 Find All Routes Between Two Cities

#5 Find All Routes Between Two Cities For Given Airline

- User is asked to supply the Arrival City (3 letter code), Departure City (3 letter code), and airline name
- If there is a flight between the cities the path is presented
- If there is a flight path with one connection between the cities both flights are shown

Team 8: Dāda Bäce
Adam Nash awn10@pitt.edu
Jason Henriquez jth79@pitt.edu
Ethan Dewit erd56@pitt.edu

#6 Find All Routes With Available Seats Between Two Cities on Given Date

- User is asked to supply the Arrival City (3 letter code), Departure City (3 letter code) and Departure Date (Formatted as MM-DD-YYYY)
- If a flight exists that isn't full between the two cities, it will be shows
- If a flight path exists with one connection between the two cities without either of the flights being full, it will show

#7 Add reservation

- Prompts the user for their First Name and Last name and checks if they are in the system
- Requests flight number for each leg of the trip
- Prompts the user for the date of their flight
- Calculates the price of the reservation
- Checks if flight is full and confirms flight if not full

#8 Delete Reservation

- User is prompted to supply the reservation number to be deleted
- If the specified reservation exists, all of the associated information is deleted (all legs)
- The flight is switched to a smaller capacity plane if the updated number of passengers can fit in a smaller capacity plane

#9 Show Reservation Information For Given Reservation Number

- User is prompted to to supply reservation number
- If reservation exists, all flights connected to the given reservation number will be displayed
- If specified reservation number does not exist, an error message is displayed

#10 Buy Ticket From Existing Reservation

- User is prompted to supply a reservation number
- If reservation exists, the reservation is now marked as "true" for being ticketed

#11 Find the Top-k Paying Customers for Each Airline

- User is prompted to supply k, the number of top-k customers they wish to display
- System will display, for each airline, the top-k customers who have payed the most to each airline

#12 Find the Top-k Traveled Customers For Each Airline

- User is prompted to supply k, the number of top-k customers they wish to display

Team 8: Dāda Bäce
Adam Nash awn10@pitt.edu
Jason Henriquez jth79@pitt.edu
Ethan Dewit erd56@pitt.edu

- System will display, for each airline, the top-k customers with the highest number of legs with that airline. In case of tie, each tying customer and amount is shown

#13 Rank the Airlines Based on Customer Satisfaction
- The airlines are ranked on the number of customers that bought tickets on each airline
- The rankings are then displayed to the user

Administrator Interface Features

#1 Erase Database
- User is asked to verify that they wish to delete all the data by entering 'Y'
- If 'Y' is entered, all the tuples of all the tables in the database are deleted

#2 Load Airline Information
- User is prompted to supply a filename (path to file on local machine) where airline information is stored.
- Input is formatted like: United Airlines UAL 1931
- Each field must be separated by a tab
- This function will take each line of the input file and insert a new entry in the Airline table

#3 Load Schedule Information
- User is prompted to supply a filename (path to a file on local machine) where the flight airline schedules are stored
- Input it tab separated and requires all parameters to exist
- Adds to the database and does not replace
- Does not Handle duplicate flights being added

#4 Load Pricing Information
- User is prompted to enter 'L' to load pricing information or 'C' to change thee price of an existing flight
- If the user chooses C, the user is prompted to supply departure city, arrival city, high price and low price. The price of that flight is then updated
- If the user chooses L, the user is asked to supply the filename where the pricing information is stored
- File must be in following format: PIT JFK 1 250 120
- Attributes must be separated by a tab

Team 8: Dāda Bäce
Adam Nash awn10@pitt.edu
Jason Henriquez jth79@pitt.edu
Ethan Dewit erd56@pitt.edu

#5 Load Plane Information

- User is prompted to supply a filename (path to a file on local machine) where the plane information is stored
- Input is tab separated and requires all parameters to exist
- Adds to the database and does not replace
- -Does not Handle duplicate flights being added

#6 Generate Passenger Manifest for Specific Flight on Given Date

- User is prompted to supply the flight number and a date
- The database is then queried for the flight number and date
- If specified flight exists on given date, a list of passengers (salutation, first name, last name) is displayed

#7 Update the Current Timestamp

- User is asked to supply a date and a time
- Specified input date and time is used to update c_timestamp in OurTimestamp table

Limitations

- Financial transactions cannot be verified, credit card numbers of correct length are assumed to be valid
- Insecure, there is no validation for administrator tasks, which include deleting the whole database

Possible Future Improvements

- Implement authentication of administrators
- Implement more secure authentication of customers based on any of a few possible methods (e.g., driver's license option, SSN option, etc.)
- Implement secure processing of financial transactions
    - Ensure that credit cards are valid
    - Ensure that payment was successfully processed
- Implement more User Friendly actions
    - List all flights
    - GUI
    - Option to go back on incorrect input