Proposed Project Summary

Airline & Automobile & Hotel Reservation System

Albert Martinez
Rafael Jimenez
Mark Anthony Bedoya

ECE 373

Problem Statement

Our project will simulate a reservation system, which incorporates airline, hotel, and car rental reservations all in the same transaction. In this modern age, people have become accustomed to on-demand service. There has been a transition from customers having to wait on the phone to speak with customer service, to now relying on their mobile devices to handle scheduling. This transition has also been beneficial for industries, since they no longer need to employ a large work force to handle the large call volume. Our design will take into consideration both of these concepts. From an industry standpoint, we still want to maintain a high level of customer service and for the user we want to present a hassle-free application. Our project will have an emphasis on ease of use, by incorporating pop-up messages to determine user needs and assist them throughout the reservation transaction. We will incorporate 4 interfaces consisting of a Main Menu, Flight Reservation, Hotel Reservation, and Car Rental Reservation. The user will have the option of making reservations for all three or just one

Domain Analysis

Because each class is designed to have only a single responsibility, many classes will be ultimately used to build the entire reservation application. For example, the "Flight" portion will consist of Coach and Premium classes. The "Hotel" portion will be broken down into Single, Double, and Master Suite classes. The "AutoRental" segment will consist of Compact, Mid-Size, and Luxury classes. These are just preliminary classes, which will continue to evolve and grow as the planning stages become more in depth.

For the sake of simplicity for the proposal, the scope will be limited as if there were only four classes consisting of Passenger, Flight, Hotel, and CarRental. There are essentially two main parts for each airline, hotel, and automobile segment. There is the user management portion, which would allow users to manage their reservations. Then there is the tracking portion which consist of managing Flights/ planes, passengers/Passengers, Hotel/rooms, and Vehicle/cars. A LinkedList<Passenger> will be incorporated to track and access passengers. The LinkedList will allow for interaction between the different classes. The following is a rudimentary UML for the reservation system.

Passenger
- Name : String
- flight : Flight
- vehicle : AutoRental
- hotel : Hotel
+ Passenger()
+ getName() : String
+ setName(String Name) : void
+ addFlight(Flight flight) : void
+ getFlight() : flight
+ addHotelRoom(Hotel hotel) : void
+ getHotelRoom() : hotel
+ addCarRental(AutoRental vehicle) : void
+getCarRental(): vehicle

Flight
- Name : string
- FlightNumber : Integer
- schedule : Integer[ ]
- flight_roster : Passenger[ ]
+ Flight()
+ getName() : String
+ setName(String Name) : void
+ getFlightNumber() : Integer
+ setFlightNumber() : void
+ getSchedule() : Integer[ ]
+ setSchedule(Integer schedule_input) : void
+ addPassenger(Passenger passenger) : void
+ getFlighttRoster() : Passenger[ ]

Hotel
- Name : String
- RoomNumber : Integer
- roomroster : Passenger [ ]
+ Hotel()
+ getHotelName() : String
+ setHotelName(String Name) : void
+getRoomNumber():Integer
+setRoomNumber(Integer RoomNumber):void
+ addPassenger(Passenger passenger) : void
+ getRoomRoster() : Passenger[ ]

CarRental
- Model : String
-Make : Sring
-Type : String
-passenger : Passenger[]
+ CarRental()
+ getModel() : String
+ setModel(String Model) : void
+ getMake() : String
+ setMake(String Make) : void
+ getType() : String
+ setType(String Type) : void
+ addPassenger(Passenger passenger) : void
+ getPassenger() : Passenger[ ]

Interaction Analysis

The following diagram (*fig 1*) depicts the general flow of the application. The user will initially be taken to a *Main Menu*. Depending on their selection, the user will be taken to the appropriate interface to either begin making an entry or to modify a previous entry.

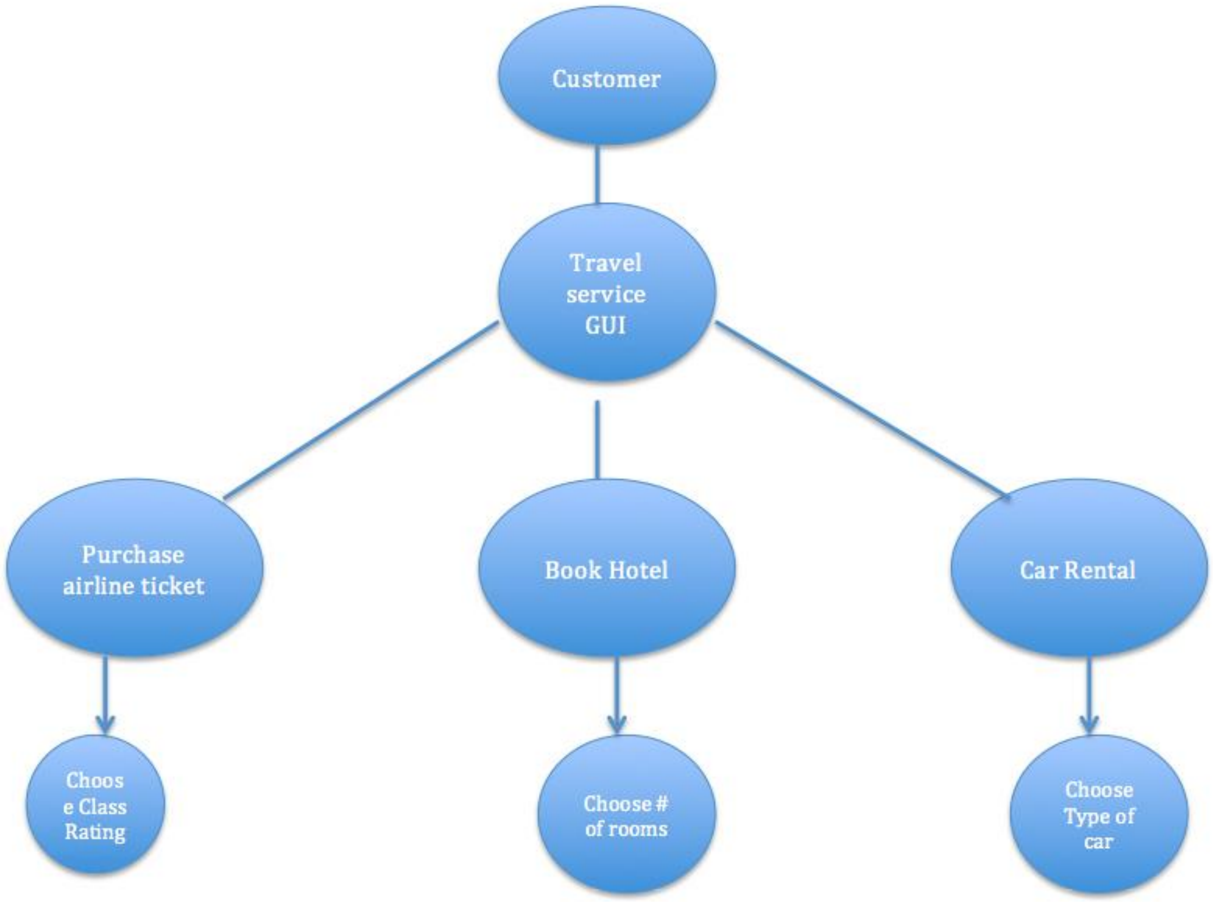
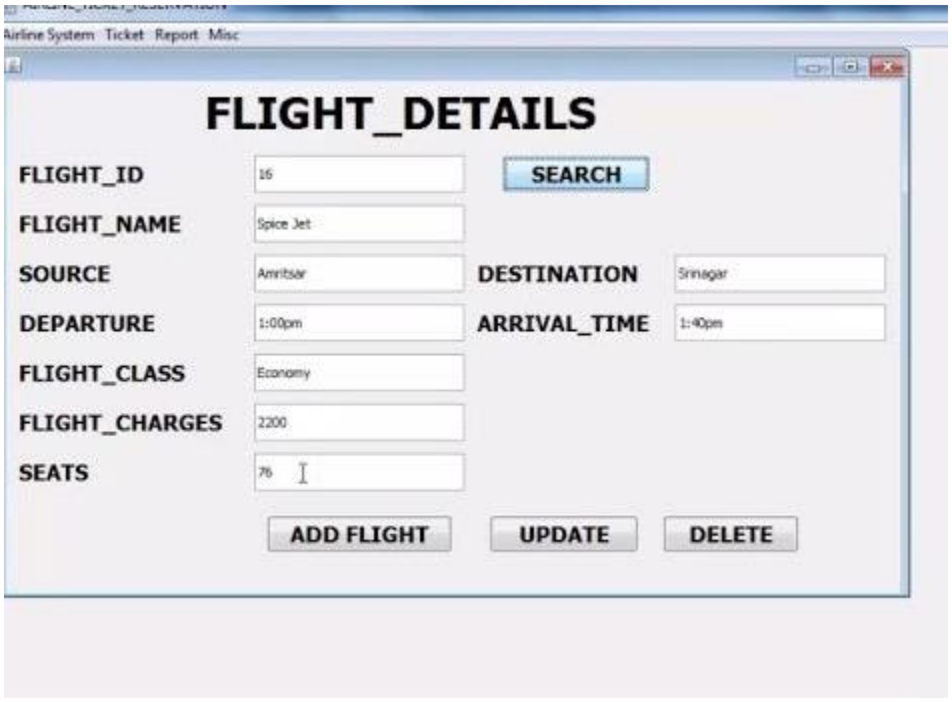


Fig 1.

For the user interface, our goal will be to design a friendly and easy to use GUI. The final result will be a series of pull down menus, and text fields, along with pop-up messages to help guide the user throughout the process. We will incorporate 4 interfaces consisting of a Main Menu, Flight Reservation, Hotel Reservation, and Car Rental Reservation. The user will have the option of making reservations for all three or just one. During the initial planning stages, the group has been researching examples online to get a general idea on how to incorporate all the desired features. The following diagram (*fig 2*) depicts a general idea of how one of our interfaces would look like. We would like to stick with the simple design but at the same time, incorporate our own personal touches to enhance the user's experience.



FLIGHT_ID	16		
FLIGHT_NAME	Spice Jet		
SOURCE	Amritsar	DESTINATION	Srinagar
DEPARTURE	1:00pm	ARRIVAL_TIME	1:40pm
FLIGHT_CLASS	Economy		
FLIGHT_CHARGES	2200		
SEATS	76		

Fig 2.

Plan of Work

Considering that our project consist of 3 major portions, each team member will be responsible for one of the portions. Unlike other projects where the work gets divided and then put together at the end, this will not be a successful strategy for this team. When working with Object Oriented Design, there are core principles that need to be incorporated such as Abstraction, Encapsulation, Polymorphism, and Inheritance. Therefore, in order to create a clean and modular design, it will be important for the team to continuously collaborate with each other and run several tests during the entire build process.