Airline Ticket Booking System

Dylan Herthoge, Jonathon Mangan, Phoebe Spratt

Group 34

Changes

Overall, we didn’t change our design too drastically. The classes that we designed in our diagram stayed the same, but the storage of objects was overhauled after we realized using text files would be tedious. Besides that, most of our changes were minor: adding and removing methods to implement our code properly. Below is a list describing most of the changes from our initial design.

* Moved to serialization for persistence instead of reading and writing to text files
  + Did it to simplify
  + Added Serializable to the UML diagram
* Added methods to the review menus for readability
* Added two exception types to handle people/airlines registering under the same username
* Added the distance between the source and destination to flight to give passengers miles to spend on move flights if the flight was canceled
* Didn’t utilize PassengerFlightNeeds class, just needed to include a constructor in Ticket that could add a String for meal preference
* PassengerMenu had to have multiple variables added including database, scanner, airlines, sort, and filters
* PassengerMenu had to have multiple methods added including makeTicket, checkOutOfDateTickets, checkCompletelyBooked, checkFlightHasLeft, and removeBlackListAirlines
* PassengerMenu removed the methods logout(), and search() as they were implemented within the startMenu() method
* Added methods to get flights by their flight number
* Moves the flight type instance variables from to the flight class
* Changed the majority of the old sets into ArrayLists
* Added a createFlight method into the airline menu

JUnit Testing

* Tested methods that sort/filter, but found it hard to test methods that normally use scanner input
* Our driver that controls most of the program is MenuDriver
  + There are also drivers for PassengerReviewMenu and AirlineReviewMenu
* More Testing info can be found in the README.txt

UML Diagram