Airline Reservation Term Project

15 December 2015

CIS 3270

By

Javar Reid

Giorgi Sukhitashvili

Rodney Barton

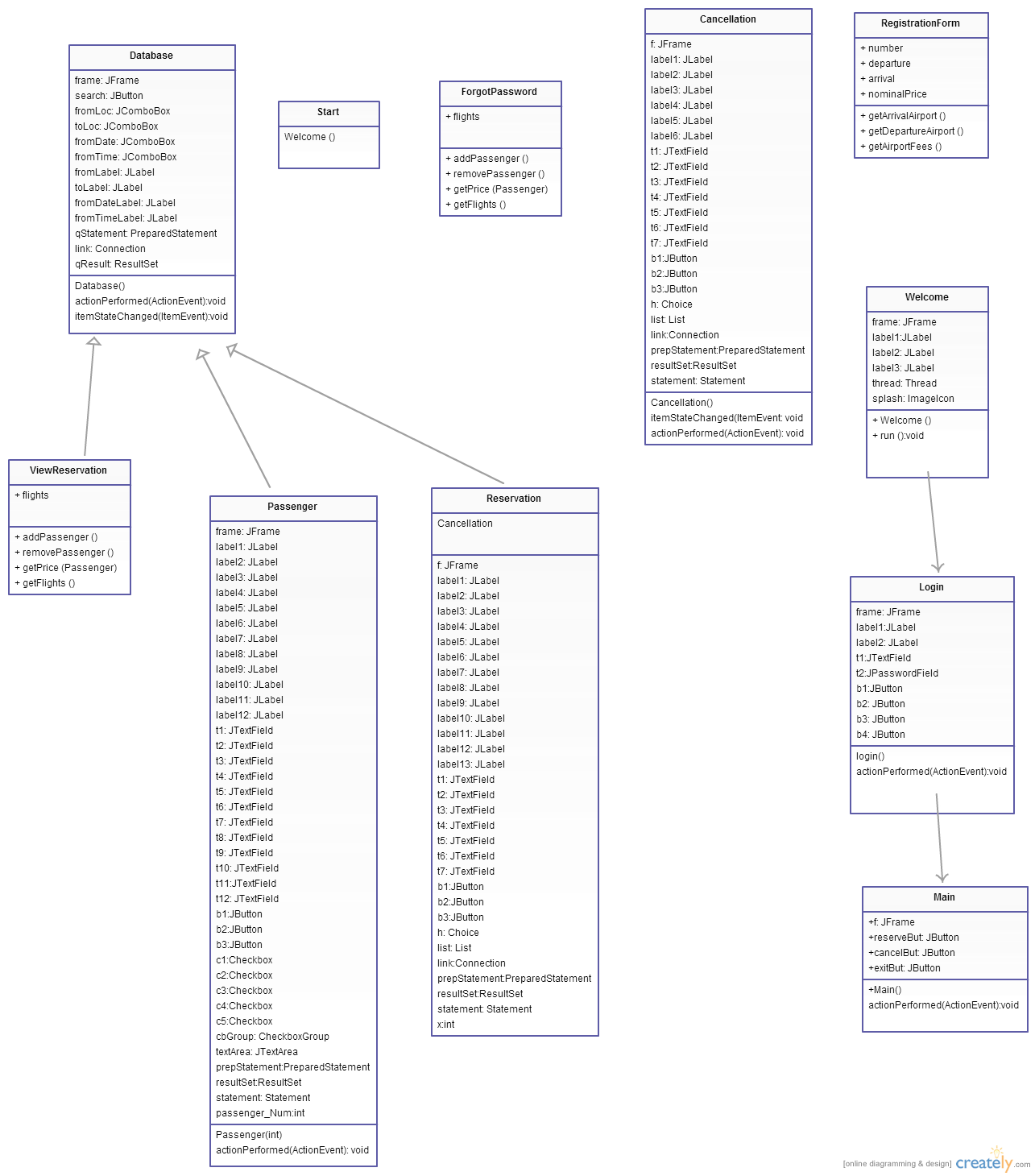
**Requirements and Specifications:**

We were assigned to develop an airline reservation system using Java. The system should begin with a splash screen, then a menu with the option to login with username/password or register. Registration requires indicative identifying information such as, name, address, social security, email, telephone, etc. The user should be able to retrieve password by answering a security question if they forgotten it. There should be two types of users: admin (who has more privileges) and customer. A customer should be able to search, add, and delete flights from account, but should not be able to add flight more than once.

The application should warn users about conflicting flight times, and should not allow users to overbook flights. User should be able to log out and log back in and their information should stay intact. Admins should be able to do everything a customer can do, as well as being able to add, update, or delete flights. The application should always provide an option to go back to the main menu.

We were also required to have at least four different classes, use at least three concepts of Encapsulation, Inheritance, Polymorphism, or Composition, and classes with Abstract and Interface. Our code should have comments, and should use exception handling to prevent it from crashing. Our project also involved database programming (JDBC), and we chose MySQL as our database of choice.

**Class Diagram:**



**Data Model:**

* Passenger (**Passenger\_Num**, Username, Password, First\_Name, Last\_Name, Social\_Security\_Num, Address, Zip, State, Telephone, Email, Security\_Question, Security\_Answer)
* TempPassenger (**Passenger\_Num**, Username, Password, First\_Name, Last\_Name, Social\_Security\_Num, Address, Zip, State, Telephone, Email, Security\_Question, Security\_Answer)
* Flights (**Flight\_Num**, From\_Loc, To\_Loc, Depart\_Date, Depart\_Time, Arrival\_Time, Total\_Seats)
* Reservation (**Reservation\_Num**, Passenger\_Num, Flight\_Num)

**Functionality and Flow:**

* Start class - The program begins in the Start class. The Start class is a basic class that holds the Main method. In the Main method, there is a call to the default constructor for the Welcome object of the Welcome class.
* Welcome class - implements the Runnable interface. A splash screen comes on. The frame runs on the screen for a few seconds before the Login object is created.
* Login class – implements actionListener. A frame appears and prompts the user to enter username and password, with text fields, and two buttons: Forgot Password, Ok, Registration, and Cancel. If the user selects Ok without any input, the system tells them that fields are blank. If the user inputs the wrong username or password, the system will notify them. At the moment, our program only allows one username and password (admin; 1234). Once the correct username and password is input, the program opens the Main object, which opens the main menu.
* RegistrationForm class – allows the user to register their account by submitting identifiable data.
* ForgotPassword – allows the user to enter the username and security question to retrieve their password.
* Main class – implements actionListener. Allows the user to go to Reservations, Cancellations, or Logout.
* Reservation class – Reservations allows user to select a flight.
* Cancellation class – Allows user to cancel flights.