**Programmer’s Manual**

*Packages:*

1. Javas
   1. User.Java
   2. DBClass.java
   3. Logout.java
   4. Bean.java
   5. Print.java
   6. BookedFlight.java
2. jinfeng.myservlets
   1. login.java
   2. FlightSearchResult.java
   3. TransactionConfirmation.java
3. servelets
   1. Registration.java
   2. SearchQuery.java
   3. TravelPackages.java
   4. TravelPackagesServlet.java
   5. ViewAndBook.java

**User.java**

* Users() - Constructor of Users class. It creates an Users object and connections to the database.
* Users(String1, String2, String3, String4) - Constructor of Users class. It creates an Users object and connections to the database. It also reads the user information from arguments.
* Users(String1, String2) - Constructor of Users class. It creates an Users object and connections to the database. It also reads the user information from arguments. Also the method checks whether the user is logged in.
* insert() - Add the detailed user information to the database
* closeConnection() - Close database connection.
* getLoggedIn() - check whether the user is logged in or not.
* check() - check whether the username and password matches that in the database. Then decide whether the user logs in successfully or not.
* checkUsername() - check whether a username exists in the database.
* setUsername(String)
* getUsername()
* setPassword(String)
* getPassword()
* setEmail(String)
* getEmail()
* setDob(String)
* getDob()

**DBClass.java**

* DBclass() - Constructor. Use default database settings.
* DBClass(String1, String2, String3, String4, int, String5) - Set database settings including connName, username, password, hostname, port number, SID.
* connectMeIn() - Connect to the database
* ResultSet execute(String) - Execute a SQL query, and store the selected information in the ResultSet.
* insert(String) - Execute a SQL query.
* closeConnection() - Close the database connection

**login.java**

* login()
* doGet(HTTPRequest, HTTPResponse) - method that accepts http requests/responses from the login.jsp and calls the doPost method using the same parameters.
* doPost(HTTPRequest, HTTPResponse) - method accepts http requests/responses from the login.jsp. It then utilizes the input from the *usrname* and *pass* textboxes from the login.jsp webpage. If the username and password match the data in the database, the program would let the user login in and create a session for the user. Then the program will direct user to the SearchQuery.jsp. If the input does not match data in the database, the program would inform the user and let the user retype the username and password.

**Registration.java**

* **registration()** -constructor that extends the httpservlet class.
* doGet(HTTPRequest, HTTPResponse) - method that accepts http requests/responses from the login.jsp and calls the doPost method using the same parameters.
* doPost(HTTPRequest, HTTPResponse) - method accepts http requests/responses from the registration.jsp. The method get input of registration information from HTML fore in the registration.jsp. Then the program would check whether the username exists or not. If not, the program would let the user register successfully and insert the user information to the database.

**SearchQuery.java**

* **SearchQuery()** -constructor that extends the httpservlet class.
* doGet(HTTPRequest, HTTPResponse) - method that accepts http requests/responses from the SearchQuery.jsp and calls the doPost method using the same parameters.
* doPost(HTTPRequest, HTTPResponse) - method accepts http requests/responses from the SeaerchQuery.jsp. The program would search in the database based on the input information of the user. If there is flight which matches the input, the program will store the information of those flight in a ResultSet. Then the program passes the ResultSet to the SearchResult.jsp.

**SearchResult.java**

* **SearchResult()** -constructor that extends the httpservlet class.
* doGet(HTTPRequest, HTTPResponse) - method that accepts http requests/responses from the SearchResult.jsp and calls the doPost method using the same parameters.
* doPost(HTTPRequest, HTTPResponse) - method accepts http requests/responses from the SeaerchResult.jsp. The program would get the flightID of the flight the user has choosed. Then the program retrieve the information of the flight and direct user to the ViewandBook.jsp.

**ViewAndBook.java**

* **ViewAndBook()** -constructor that extends the httpservlet class.
* doGet(HTTPRequest, HTTPResponse) - method that accepts http requests/responses from the ViewAndBook.jsp and calls the doPost method using the same parameters.
* doPost(HTTPRequest, HTTPResponse) - method accepts http requests/responses from the ViewAndBook.jsp. The program would get the flight parameters passed by ViewAndBook jsp page. It will verifies the availability of the requested number of seats. The program then would redirect the user the the transaction.jsp with the flight details, number of seats and total cost.

**TransactionConfirmation.java**

* **TransactionConfrimation()** -constructor that extends the httpservlet class.
* doGet(HTTPRequest, HTTPResponse) - method that accepts http requests/responses from the Transaction.jsp and calls the doPost method using the same parameters.
* doPost(HTTPRequest, HTTPResponse) - method accepts http requests/responses from the Transaction.jsp. The class get banking information and total cost from Transaction page.

**BookedFlight.Java**

* BookingFlight is an object used to store the Flight Number, cost, and amount of seats wanted for a certain flight. This in combination with java’s List utility allows for persisten storage to be passed throughout the web page.
* **BookingFlight()** is the default constructor. It takes three parameters an integer flight number, an integer seats wanted, and a double cost.
* getFlightNumber - returns the object’s flight number.
* getWantedSeats - returns the object’s wanted seats.
* getDestination - returns the object’s destination
* getDescription - return the object’s description
* getPackageid - return the object’s package id
* getCost - returns the total cost of the flight order.

**Logout.java**

* Logout is a servlet used in the search query.jsp. This is used to allow users to logout from the program. This allows code to be embedded on all JSP pages to check if a user is logged in or not.
* doPost(HTTPRequest, HTTPResponse) clears the BookingFlight list and updates the session logged in attribute accordingly. Then it sends the user to the login page.

**Print.java**

* doPost - get the parameter about passenger’s information from previous page and stored in with Bean. Then pass it to print.jsp page.

**TravelPackages.java**

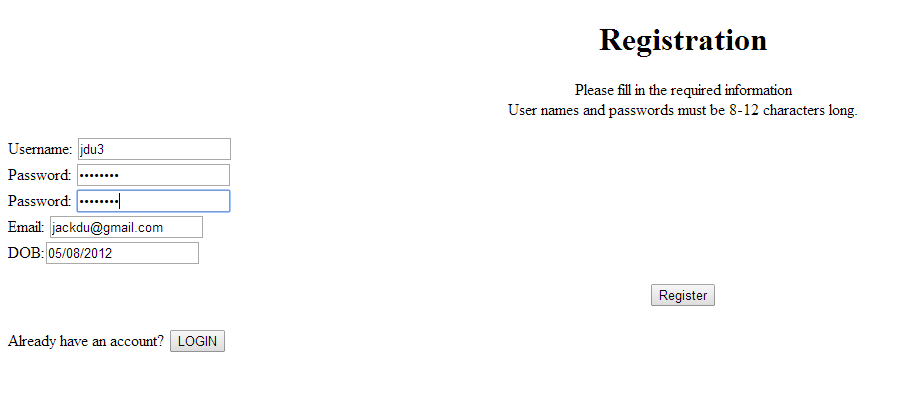
* doPost - get the package from the database based on the information of the flight the user choose. Then store the package information in the session. Finally direct the page to travelpackages.jsp page

**TravelPackagesServlet.java**

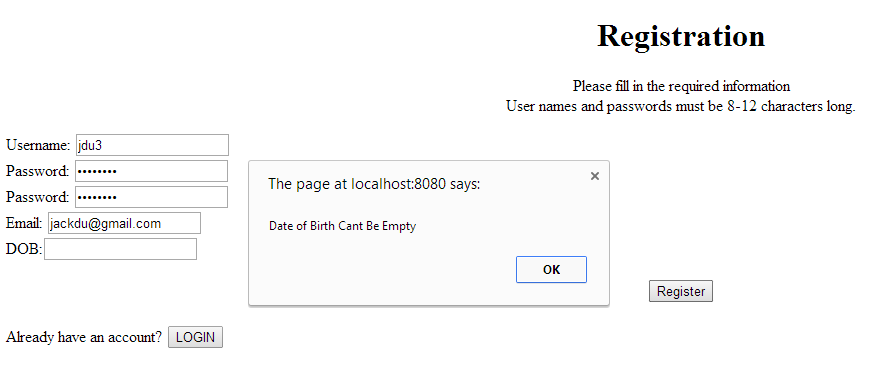
* doPost - get the package information the user has just order and add it to the shopping cart.

**User’s Manual**

1. Registration (data should be in appropriate form)

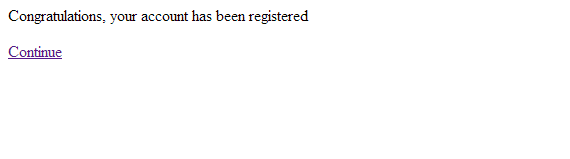


If not in appropriate form, the program will alert you.





2. If register successfully

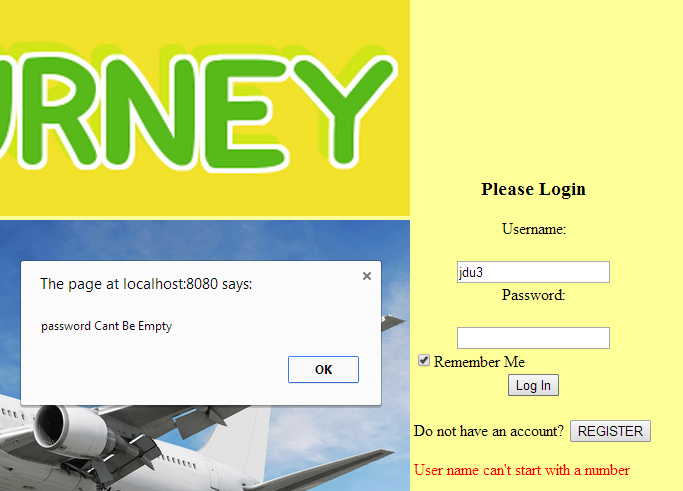


click “continue ” to login

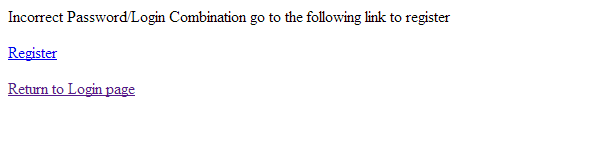
3. login. Choose whether to remember username or not



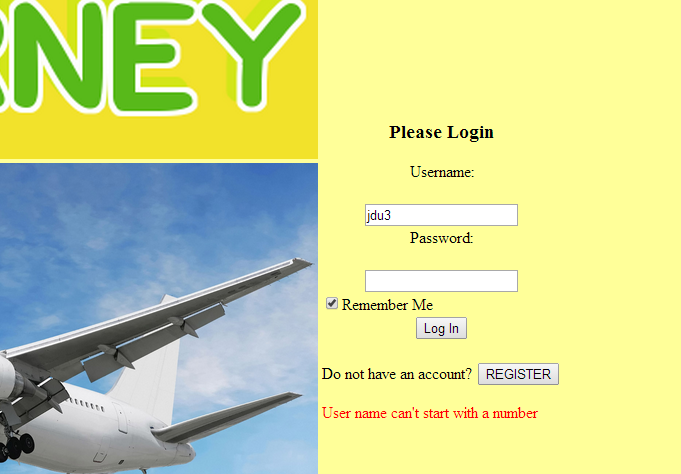
If the input if not proper?



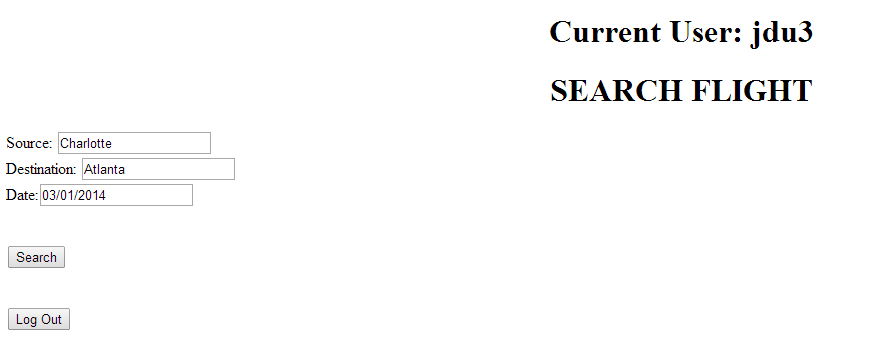
Wrong password?



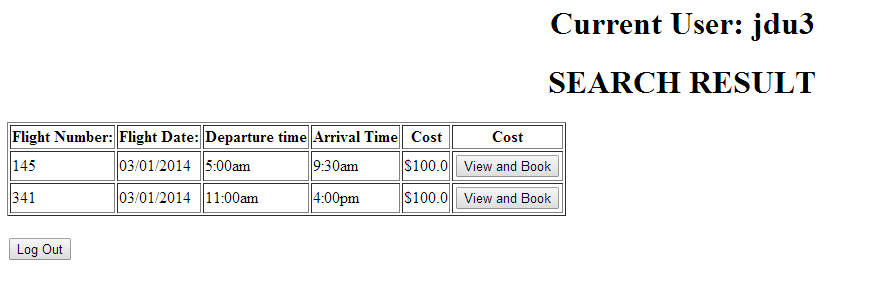
if you choose to remember the password, next time, when you visit the login page, your username would be presented.



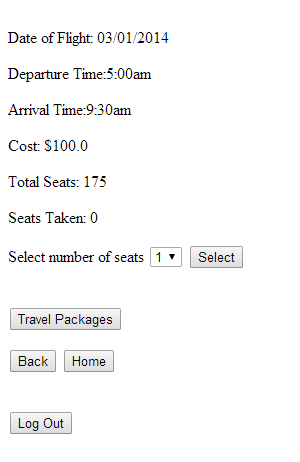
4. type in the flight you want to search



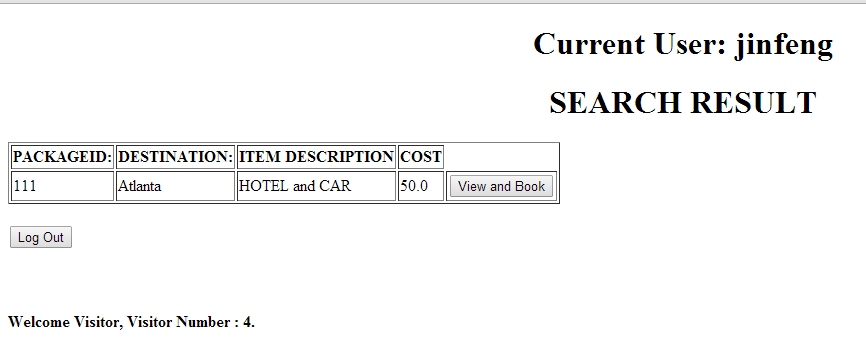
5. Choose the flight you want to order.



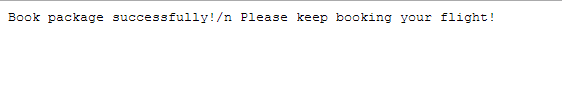
6. Directed to view and book page



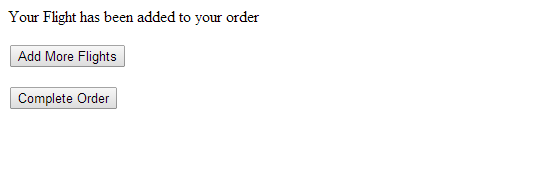
7. Choose travel packages if you want to buy the package for car rental service and hotel service



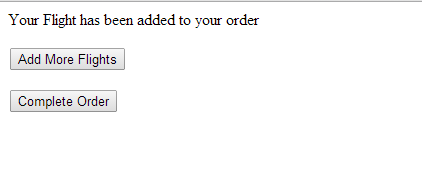
Add package to the shopping cart



8. Go back to view and book page to keep booking flight by selecting “select”

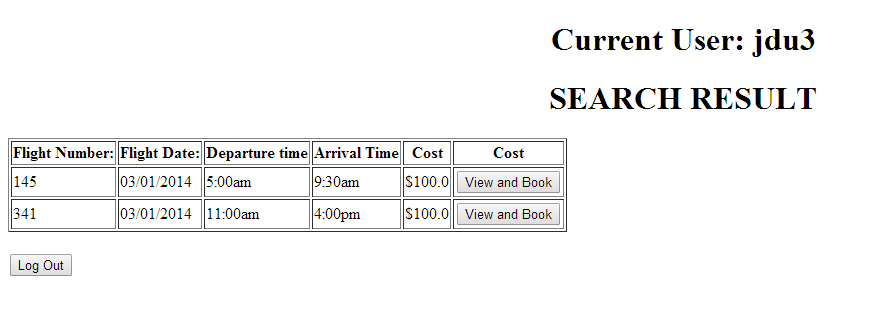


9. Want to add another item in the shopping cart?

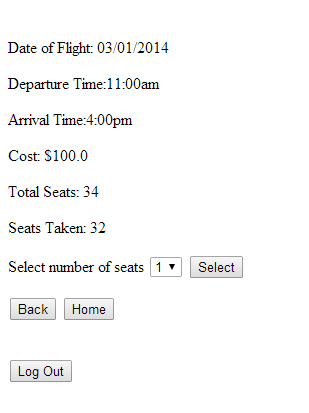


10. Add more flights. And repeat the former procedure.

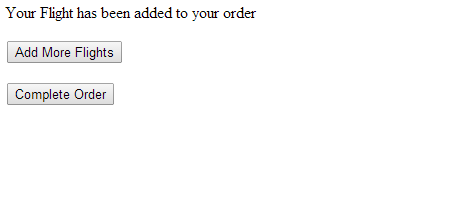




After View and Book another item

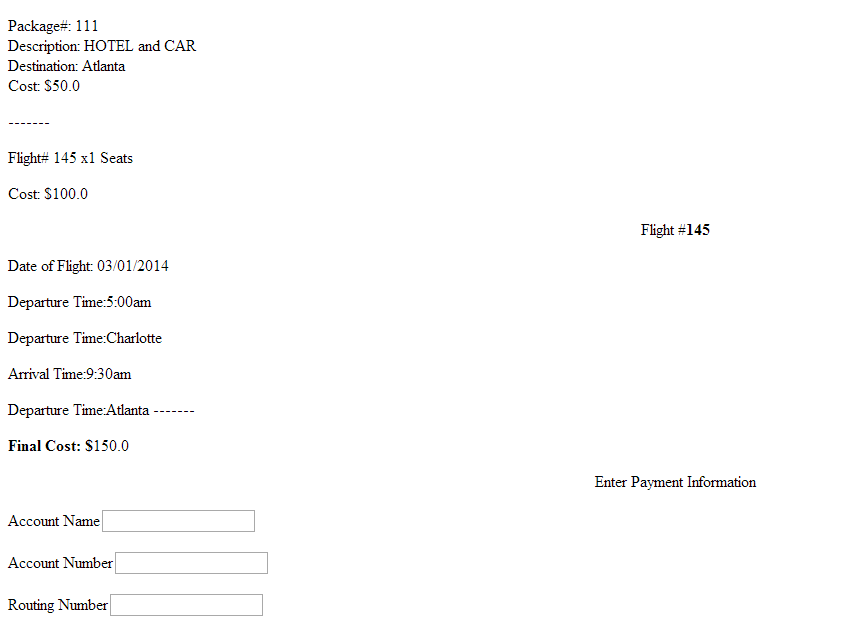


Select

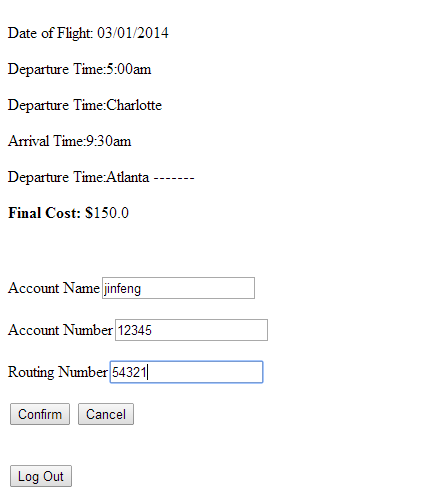


11 Complete your order.

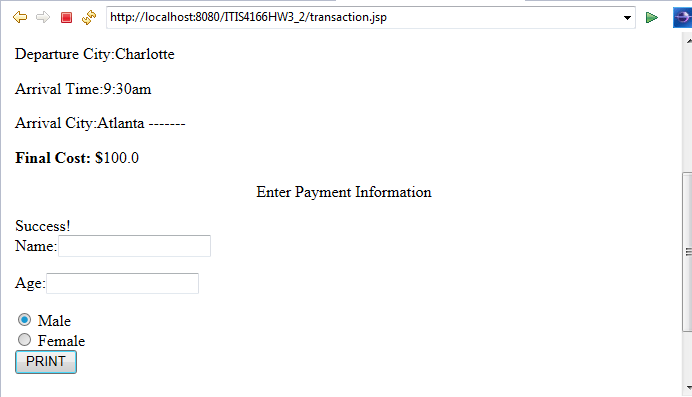
Flight information, package information and cost are displayed. Input the bank account info.



11. Type in the payment information and confirm

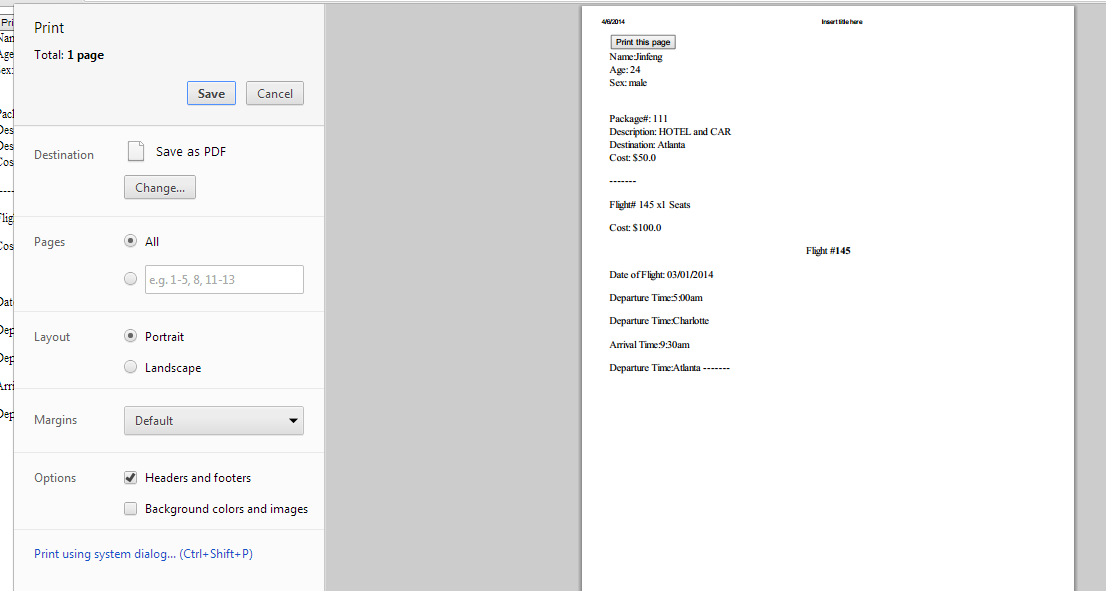


12. Purchase confirmed. Then input the information of the passenger then click print



12 Click print to print the page





13. Log out.

