JEE MINI PROJECT

JEE
Mini Project
Airline Reservation System
(ARS)

Document Control

Reference No:			
Security Classification:	**Internal**		
Issue Date:			
Author(s):	**Name**		**Title**
	Shrilata Tavargeri		Manager – T & D
Reviewer(s):	**Name**		**Title**
	Training Team		Executive, Managers – T&D
Issuer(s):	**Name**		**Title**
	Dinesh Anantwar		Sr. Manager
Issuer Signature(s):			
Distribution:	**Name**		**Title**
	Team		All Designations in Training
Document History:	**Date**	**Revision**	**Change**
	Nov 24, 08	0.01D	Initial draft
	Nov 29, 08	0.02D	Review comments incorporated
	July 03,13	0.03D	Changes in Miniproject document content to standardize for 60 days training.
	June 29,15	0.04D	Changes in Miniproject document content as per the upgraded courses

Table of Contents

1 INTRODUCTION

This document outlines a mini project for the J2EE LOT. The project is to develop an Airline Reservation System (ARS). This document contains the work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the J2EE LOT.

1.1 SETUP CHECKLIST FOR MINI PROJECT

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP, Windows 7
- Memory: 32MB of RAM (64MB or more recommended)
- Internet Explorer 6.0 or higher
- Oracle 9i client and access to oracle 9i server
- JDK 8
- Eclipse Luna
- JUnit 4.0

1.2 INSTRUCTIONS

- The code modules in the mini project should follow all the coding standards.
- Create a directory by your name in drive **<drive>**. In this directory, create a subdirectory **MiniProject**. Store your Project here.
- You can refer to your course material.
- You may also look up the help provided in the java docs The total time required to complete this mini project is 50 hrs.
- Since this project work will span over couple of months, you will need to take care of maintaining the code

2 PROBLEM STATEMENT

2.1 OBJECTIVE

Development of an Airline Reservation System (ARS)

2.2 ABSTRACT OF THE PROJECT

This project is aimed at developing an Airline Reservation System (ARS) for customers. This system can be used to search flight details, reserve flight, and update/cancel any reservation. This is an integrated system that contains both the user component and the administration component. There are features like report generators etc in this system.

2.3 FUNCTIONAL COMPONENTS OF THE PROJECT

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate; you can make appropriate assumptions and proceed.

There are three categories of people who would access the system viz. customer, airline-executive and administrator. Each one of them would have some exclusive privileges (for e.g. User can reserve a flight ticket and also update or cancel his reservation, Airline-executive alone will be able to view flight-occupancy and only the administrator has the right to keep track of the flight details.)

1. User should be able to
   - Place a request for a flight ticket.
   - User should be able to view the reservation details, update and cancel the booking.
2. The airline-executive should be able to :
   - Login into the system using his/her credentials.
   - View flight-occupancy details.
3. The administration should be able to
   - Login to the system using his/her credentials
   - Update and manage flight schedules.

- Update and manage flight information
- Generate various reports like:
    - View List of flights on a particular day, to a particular destination etc
    - View Bookings of specific flight
    - View passenger list of specific flight

2.4 TECHNOLOGY USED

- *Business Logic Components and Services :-*
    1. Java Beans
    2. Java component classes

- *Databases:-*
    1. Oracle 9i

3 Implementation in JEE LOT

3.1 Summary of the functionality to be built:

The participants need to develop the **ARS** by building the functionality incrementally in each of the course modules of JEE LOT.

Sr. No	Course	No. of Saturdays	Functionality to be built
1	Programming Foundation with Pseudo code	~~1~~	Analyze the given case study
2	Web Basics (HTML 5,CSS 3, JavaScript, XML)	~~1~~	Analyze the given case study ~~Create HTML Pages for your Project~~
3	Oracle Basics	1	Creating relevant database tables
4	OOP & UML	1	Creating relevant Use case and class diagrams
5	Programming Foundation with Pseudo code + Web Basics +Oracle Basics +OOP & UML Test		
6	Core Java 8 & Development Tools (Junit, Log4j)	2	Developing Business components (java classes). Coding for test classes & testing the functionality using JUnit
7	Core Java 8 + Dev Tools + OOP/UML Test		
8	Servlets	2	Developing the application using the prototypes. Creating Java classes and implementing code for displaying data. Integration of application.
9	JSP		
10	Developer Workbench (PMD, MAVEN)		
11	Servlets + JSP + Dev Workbench Test		
12	Basic Spring 4.0	1	Prepare document for presentation.
13	Basic Spring Test		
14	Mini Project presentation		

Formatted Table

3.2 Guidelines on the functionality to be built:

The functionality and components to be built in each of the course modules of J2EE LOT is as follows: Convert your ~~HTML~~ Input Pages into Console Screen using Java classes. (One java class for one HTML page design)

1. **Course: ~~HTML, JavaScript~~Core Java (Duration: 10 hours)**
   a. Develop the following screens:
      i. View Flight Schedules screen: This functionality provides the user with information related to flights scheduled by Airline Company.
      ii. Make a reservation: Allows the user to reserve tickets using the ARS.
      iii. View or change the booking screen: For users, the customer should be able to view the booking or change the booking.
      iv. Maintain Flight Information: This functionality should be available only for authorized administrators. Administrations maintain information includes Flight number, Origin, Destination, Departure Time, arrival Time, Days on which the flight is available. Apart from this the, overall capacity available for reservations on the flights is also maintained. There could be multiple flights available between any 2 cities on the same day. They are identified on the basis of the flight numbers.
      v. View Flight Occupancy Status Screen: This screen should only be available to authorized Airline Executives. He must be able to see the occupancy based on following parameters:
         1. Occupancy from a given flight for a given period.
         2. Overall Occupancy from all flights flying from a particular source to a particular destination.
      vi. Other Miscellaneous Information: Provide company's call centers contact numbers, FAQs, Terms and conditions and a feedback form.

In this course you need to develop the user interface using java classes and document the flow of your application including the screen shot of console in a word document. The screens should include the fields as per the functionality mentioned above

2. **Course: Oracle (Duration: 5 hours)**
   a. Create the following database tables:
      i. Users: This will contain the list of valid users (airline-exec and administrator only).
      ii. Airport: This will contain details of every airport
      iii. FlightInformation: This will contain details of every flight scheduled by this airline.

iv. BookingInformation: This will contain details of booking/reservation done by passengers.

b. The structure of the above listed tables is as follows:

i. Users: username(varchar2(20)), password (varchar2(20)), role(varchar2(10)), mobile_no(number)

ii. Airport: AirportName (varchar2(20)), Abbreviation (varchar2(5)), Location (varchar2(40))

Note: Location can be normalized since it would contain further information like city, state, zip code etc.

iii. FlightInformation: flightno(varchar2(5)), airline (varchar2(10)), dep_city (varchar2(10)), arr_city (varchar2(10)), dep_date (date), arr_date (date), dep_time, arr_time, FirstSeats (number), FirstSeatFare (number(m,n)), BussSeats (number), BussSeatsFare (number(m,n))

iv. BookingInformation: Booking_id (varchar2(5)), cust_email (varchar2(20)), no_of_passengers (number), class_type (varchar2(10)), total_fare (number(m,n)), seat_number(s), CreditCard_info (varchar2(10)), src_city (varchar2(10)), dest_city (varchar2(10))

3. **Course: OOP & UML (Duration: 5 hours)**
   a. Develop relevant Use case and Class diagrams for the **ARS** application.

4. **Course: Core Java 8 + Developer Tools (Duration: 14 hours)**
   a. Develop business components (java classes) for the following functionality:
      i. User verification: This component will verify if the user who is trying to access the system is a valid user. This verification is as against the valid users listed in the users table.
      ii. Retrieve Flight Schedules: This component provides the user with information related to flights scheduled by Airline Company.
      iii. Make a reservation: Allows the valid user to reserve tickets using the ARS.

- iv. View or change the booking: The customer should be able to view the booking or change the booking.
- v. Maintain Flight Information: This is a typical CRUD (Create/Read/Update/Delete) operation. This functionality should be available only for authorized administrators. Administrators maintain information includes Flight Number, Origin, Destination, Departure Time, arrival Time, Days on which the flight is available. Apart from this the, overall capacity available for reservations on the flights is also maintained. There could be multiple flights available between any 2 cities on the same day. They are identified on the basis of the flight numbers.
- vi. View Flight Occupancy Status Screen: This screen should only be available to authorized Airline Executives. He must be able to see the occupancy based on following parameters:
  - 1.) Occupancy from a given flight for a given period.
  - 2.) Overall Occupancy from all flights flying from a particular source to a particular destination.

- b. Develop test classes for testing the following functionality
  - i. Login
  - ii. Ticket Booking.
  - iii. View Flight Schedule.
- c. Test the application using JUnit 4.0.
- d. Configure Logger to log the status of an application

5. Course: Core Java Classes + Developer Workbench (Duration: 14 hours)

- a. Convert all the java classes (business components) created in Java module to Java beans
- b. Integrate all screens(java classes created for displaying)with business components (java beans) to complete the entire functionality
- c. Configure the DataSource and modify the data access classes to use DataSource object configured.
- ~~Use https for security throughout the pages so that the valid users can only access the **ARS.**~~

6. Documentation (Duration: 2 hours)

- a. Project Documentation: Document your project details (Duration: 1 hour 30 mins).
- b. Project submission: Submit your project with all the artifacts including the test cases & documentation (Duration: 30 mins).

3.3 EVALUATION AND ASSESSMENT PARAMETERS:

This miniproject will be done in groups of five/six. Each group will identify a Team Lead who will decide which team member will code for which functionality. This project shall be evaluated at the end of spring module.

Evaluation Criteria (out of 100):

Look ~~of Application~~and Feel	**~~05~~10**
Client-side ~~and server-side ~~validation	**15~~5~~**
Code Documentation and using coding standards	**15~~20~~**
~~Overall Business logic. This includes:~~	**40**
~~• Usage of Logging API (log4j)~~Functionality	
D~~Good amount of appropriate d~~ataset~~ to showcase project completely~~	**10**
Appropriate test cases using JUnit 4.0; logging application using Log4j	**10**

Formatted: No bullets or numbering