JEE
Mini Project
Employee Maintenance System

Document Control

Reference No:			
Security Classification:	**Internal**		
Issue Date:			
Author(s):	**Name**		**Title**
	Rathnajothi		Senior Executive – T & D
Reviewer(s):	**Name**		**Title**
	Training Team		
Issuer(s):	**Name**		**Title**
Issuer Signature(s):			
Distribution:	**Name**		**Title**
	Team		All Designations in Training
Document History:	**Date**	**Revision**	**Change**
	23-07-2014	0.01D	Initial draft
	29-06-2015	0.02D	Changes in Miniproject document content as per the upgraded courses
	28-06-2016	0.03D	Changes in Miniproject document content as per the upgraded courses

Table of Contents

1 INTRODUCTION

This document outlines a mini project for the JEE LOT. The project is to develop an Employee Maintenance System for an organization. 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 JEE 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)
- 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 Employee Maintenance System used for maintenance of an employee details in an organization.

2.2 ABSTRACT OF THE PROJECT

This project is aimed at developing Employee Maintenance System. This is an Intranet based application that can be accessed throughout the organization This system can be used to search for Employees based on search condition, add individual employee, modify an existing employee details and display all employee details across locations within an organization. This is an integrated system that contains both the user (Employee) component and the Admin component

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 two categories of people who would access the system viz. employee & Admin. Each one of them would have some exclusive privileges (for e.g. Employees can just search an employee details based on search condition.)

1. Admin should be able to

    - Login to the system using his/her credentials
    - Add individual employee details by accepting all the field values from end user as listed below and inject the values into database table if data are valid else display an appropriate error messages.
        - ➢ Employee ID has to be a 6 digit number (with no decimals) and cannot start with a ZERO (Cannot be empty)
        - ➢ First and Last name should be alphabets only (Cannot be empty)
        - ➢ Date of joining and Date of birth need to undergo regular date validation (DD-MMM-YY) - (Cannot be empty)
        - ➢ The date of joining should be > Date of birth and the employee's age has to be >=18 and <= 58 as an eligibility criteria for employment
        - ➢ Department name values has to be populated from database
        - ➢ Grade has to be one of the following values: (Cannot be empty)
            1. M1
            2. M2
            3. M3

        4. M4
        5. M5
        6. M6
        7. M7
    - Designation can be alphabets limited to 50 characters: (Can be empty)
    - Gender has to be one of the following values: (Cannot be empty)
        1. Male
        2. Female
    - Basic should be numeric only and the basic has to be validated against a salary band for each grade
    - Marital Status can be one of the following values: (Cannot be empty)
        1. Single
        2. Married
        3. Divorced
        4. Separated
        5. Widowed
    - Home Address (Can be alphanumeric and optional)
    - Personal Contact Number (Can be alphanumeric and optional)
- Modify Employee details:
    - Based on existing employee id, display the following fields First Name, Last Name, Department, Grade, Designation, Basic Salary, Marital Status, Home Address, Personal Contact Number in an editable mode.(validation should be taken care for the change in basic salary if there is change in the grade)
- Display all employee details:
    - Employee details either display in single page or easy to use provisions (like previous, next, first, last options to be accepted from user.) should be provided to navigate between records. Display should be limited to 10 records per page
    - Display details in the below given tabular format

<<First>> <<Next>> <<Previous>> <<Last>>

<<ID>>	<First Name>	<Last Name>	<Department>	<Grade>	<Designation>

2. Employees/User should be able to

- Login to the system using his/her credentials
- Search an employee details based on any of the fields - ID, First Name, Last Name, department, Grade, Marital Status. The following fields are provided to enable the user specify the search conditions:
    - i. ID(Can provide wild card search)
    - ii. "First Name"   (Can provide wild card search)
    - iii. "Last Name"   (Can provide wild card search)
    - iv. Wild card search (? / *)
        - ? for a character
        - * for n characters
    - v. Department (Can choose one or more department at a time – Multiple choice)
    - vi. Grade (Multiple choice)
    - vii. Marital Status (Multiple choice)

The results will be filtered and displayed following the same display rules as the display all employee details requirement

2.4 TECHNOLOGY USED:

- ➢ *Front End :–*
    1. Java Classes
- ➢ *Business Logic Components and Services :-*
    1. Java Beans
- ➢ *Databases:-*
    1. Oracle 9i

3 IMPLEMENTATION IN JEE LOT

3.1 SUMMARY OF THE FUNCTIONALITY TO BE BUILT:

The participants need to develop the Employee Maintenance System by building the functionality incrementally in each of the course modules of JEE LOT.

Sr. No	Course	Duration (in PDs)	No. of Saturdays	Functionality to be built
1	Programming Foundation with Pseudo code	3	1	Analyze the given case study
2	Web Basics (HTML, JavaScript, XML)	4.5	1	.
3	Oracle Basics	4	1	Creating relevant database tables
4	OOP & UML	1.5	1	Creating relevant Use case and class diagrams
5	Programming Foundation with Pseudo code + Web Basics +Oracle Basics Test	1		
6	Core Java 8 & Development Tools(Development Tools (Junit, Log4j))	10	2	Developing Business components (java classes). Coding for test classes & testing the functionality using JUnit
7	Core Java 8 & Development Tools + Dev Tools + OOP/UML Test	1		
8	Servlets	3.5	2	Project specific work need not to be done as mini project is in Core Java
9	JSP	2		
10	Developer Workbench (PMD, MAVEN)	1		
11	Servlets + JSP + Dev Workbench Test	1		
12	Basic Spring 4.0	5	1	Prepare document for presentation.
13	Basic Spring Test	1		
14	Mini Project presentation	1		

3.2 GUIDELINES ON THE FUNCTIONALITY TO BE BUILT:

The functionality and components to be built in each of the course modules of JEE LOT is as follows: Here screen refers to the Console Screen. Convert your HTML Pages into Console Screen using Java class with below options.

Course: HTML, JavaScript (Duration: 10 hours)

a. Develop the following screens:
   i. Home page screen: Home option for the EMS which provides a options for the login.
   ii. Login Screen: Allows the valid user or admin to logon to the system and display the Main option screen
   iii. Main option screen: For User (Employee), this screen will display options for Search Employees screen only. For Admin, this screen will display options for Add Employee, Modify Employee, and Display all Employee screens.

iv. Search Employees screen: This screen allows the user to search an employee details based on search condition.
v. Add Employee Screen: This screen allows the Admin to add employee details in the system.
vi. Modify Employee Screen: This screen allows the Admin to modify employee details based on entered employee id if it is valid.
vii. Display all Employees Screen: This screen allows the Admin to display all employee details with pagination option.
viii. Logoff from the application at any point of time

b. **In this course you need to develop the user interface using java coding and document the flow of your application including the screenshots of screen in a word document**. The screens should include the fields as per the functionality mentioned above. Also, include client-side validations using regular expression in each of these screens.

2. Course: Oracle (Duration: 5 hours)

a. Create the following database tables:
i. User_Master: This will contain the list of valid users.
ii. Department: This contains the list of departments available in the organization.
iii. Employee: This will contain the details of an employee who have joined in an organization.
iv. Grade_Master: This will contain the details of grades applicable for an employee based on salary.

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

i. **User_Master:** UserId VARCHAR2(6), UserName VARCHAR2(15), UserPassword VARCHAR2(50), UserType VARCHAR2(10)
For Admin, and User (Employee), assume that the users are already added to the system.

ii. **Department:** Dept_ID int, Dept_Name VARCHAR2(50)

iii. **Employee:** Emp_ID VARCHAR2(6), Emp_First_Name VARCHAR2(25), Emp_Last_Name VARCHAR2(25), Emp_Date_of_Birth DATE, Emp_Date_of_Joining DATE, Emp_Dept_ID int, Emp_Grade VARCHAR2(2), Emp_Designation VARCHAR2(50), Emp_Basic int, Emp_Gender VARCHAR2(1), Emp_Marital_Status VARCHAR2(1), Emp_Home_Address VARCHAR2(100), Emp_Contact_Num VARCHAR2(15)

- iv. **Grade_Master:** Grade_Code VARCHAR2(2), Description VARCHAR2(10), Min_Salary int, Max_Salary int

Note: You may add/normalize/denormalize the tables if your application demands it.

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

4. **Course: Core Java 8 & Development Tools(Duration: 14 hours)**

    - a. Develop business components (java classes) for the following functionality:

        - i. Authentication Service (on Login): 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 User_Master table.
        - ii. Admin Service: This component will allow the admin to add new employee, update an existing employee details and display all employee details.
        - iii. User Service: This component will allow the User/Employee to search employee details based on search condition.

    - b. Develop test classes for testing the following functionality and Test the application using JUnit.
        - i. Login
        - ii. Add Employee details
        - iii. Modify Employee details
    - c. Configure Logger to log the status of an application

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

    - a. Convert all the java classes (business components) created in Java module to Java beans
    - b. Integrate all screens with business components (java beans) to complete the entire functionality

.

6. **Documentation (Duration: 2 hours)**

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

3.3 EVALUATION AND ASSESSMENT PARAMETERS:

This mini project will be done in groups of five. 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 console for all the screens	**05**
Client-side validation of inputs	**10**
Code Documentation and using coding standards	**10**
Overall Business logic. This includes: • Usage of Logging API (log4j)	**30**
Good amount of appropriate dataset to showcase project completely	**5**
Appropriate test cases using JUnit 4.0	**5**
Using MVC architecture and clean encapsulation of business logic in appropriate components. Judicious use of java beans.	**35**