

Asset Management System

Table of Contents

1 Introduction	2.
1.1 Setup Checklist for Mini Project	2.
1.2 Instructions	2.
2 Problem Statement	3.
2.1 Objective	3.
2.2 Abstract of the project	3.
2.3 Functional components of the project	3.
2.4 Technology used	4.
3 Implementation in JEE LOT	4.
3.1 Summary of the functionality to be built	4.
3.2 Guidelines on the functionality to be built	4.
3.3 Evaluation and assessment parameters	7.

1 INTRODUCTION

This document outlines a mini project for the JEE LOT. The project is to develop an Asset Management 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)
- Internet Explorer 6.0 or higher
- Oracle 10g
- JDK 8
- Eclipse Photon
- 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 Mini Project. Store your Project here.
- You can refer to your course material.
- You may also look up the help provided in the java docs and documentation provided
- 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 Asset Management System used for maintenance of an asset details in an organization.

2.2 ABSTRACT OF THE PROJECT

This project is aimed at developing Asset Management System. This system can be used to search for an asset based on search condition, assign a hardware asset to/from an employee based on request, insert new asset details, modify an existing asset details and display all asset allocation request details. This is an integrated system that contains both the user (Manager) 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. User (Manager) & Admin. Each one of them would have some exclusive privileges (for e.g. Managers can just raise a request for allocation/release of an asset for his/her team member on behalf of them.)

1. Managers should be able to

- Login to the system using his/her credentials
- Raise a request for allocation of asset to an employee, by filling up the asset requisition form that auto generates the asset request ID.
- View the status of request, based on the asset request ID

2. The administrators should be able to

- Login into the system using his/her credentials
- Include new asset/modify asset details from the inventory.
- Approve/Reject a request on the basis of the details of the request raised. If raised request is approved, then allocate asset to an employee and change request status as 'Allocated'.
- Generate various reports like:

➤ View List of assets available in an organization based on category:

1. Allocated Assets 2. Unallocated Assets

2.4 TECHNOLOGY USED

➤ Front End & Web Components:– 1. Core Java

➤ Business Logic Components and Services :- 1. Java Beans

➤ Databases:- 1. Oracle 10g

3 IMPLEMENTATION IN JEE LOT

3.1 SUMMARY OF THE FUNCTIONALITY TO BE BUILT:

The participants need to develop the Online AMS 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		Analyze the given case study
2	Web Basics (HTML 5,CSS 3, JavaScript, XML)		Developing prototype i.e. developing screens/web pages in HTML and client side validation in JavaScript.
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	java classes (business components) to java beans. Integrating Java classes with business components to complete the entire functionality. Building the Core Java application component using MAVEN build script.
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		

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:

1. Course: PI

Below should be the UI:

1) Login: Allows the valid user or admin to logon to the system and display the Main option screen.

2) Main option screen:

For Manager, this screen will display

- Raise a request for allocation of asset to an employee and
- View status of raised request screen by entering the empno for whom the request has been raised

For Admin, this screen will display for

- Add/Modify Asset Details screen,
- View all request raised and approve/reject request screen

3) Asset Allocation Request screen: This screen allows the manager to raise a request for allocation of asset to an employee.

4) View Status Screen: This screen allows the manager to view the status of request raised.

5) Add/Modify Asset Screen: This screen allows the Admin to add/modify asset details.

6) Display all request for an asset Screen: This screen allows the Admin to display all request raised by managers with approve/reject option. Logoff from the application at any point of time

2. Course: Oracle

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. Asset: This will contain the details of available hardware asset in an organization
- v. Asset_Allocation: This will contain the details of asset allocated for an employee.

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 (Manager), assume that the users are already added to the system.
- ii. Department: Dept_ID int, Dept_Name VARCHAR2(50)
- iii. Employee: Empno Number(6), Ename Varchar2(25), job Varchar2(50), mgr number(4), hiredate date, Dept_ID number
- iv. Asset: AssetId Number, AssetName Varchar2(25), AssetDes Varchar2(25), Quantity Number, Status Varchar2(15)
- v. Asset_Allocation: AllocationId Number, AssetId Number, Empno Number
References Employee(Empno) , Allocation_date Date, Release_date Date

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

3. Course: OOP & UML

a. Develop relevant Use case and Class diagrams for the application.

4. Course: Core Java 8 + Developer Tools

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 asset, update an existing asset details and Display all request for an asset.
- iii. User Service: This component will allow the manager to Raise a request for allocation of asset to an employee and check the status of request.

b. Develop test classes for testing the following functionality and Test the application using JUnit.

- i. Login
- ii. Add asset details
- iii. Modify asset details

c. Configure Logger to log the status of an application

5. Documentation

- a. Project Documentation: Document your project details
- b. Project submission: Submit your project with all the artifacts including the test cases & documentation

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 and Feel	10
Client-side validation	15
Code Documentation and using coding standards	15
Functionality	40
DataSet	10
Appropriate test cases using JUnit 4.0;logging application using Log4j	10