# Library Management System (LiMS)

Table of Contents

# 1 INTRODUCTION

This document outlines a mini project for the J2EE LOT. The project is to develop a desktop Library Management System (LiMS). 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 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 with Apache Tomcat.
- 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 a Library Management System (LiMS)

## 2.2 ABSTRACT OF THE PROJECT

This project is aimed at developing Library Management System (LiMS) for the college library. This is an Intranet based application that can be accessed throughout the campus. This system can be used to search for books/magazines; reserve books, and issue/return books from the library. This is an integrated system that contains both the user component and the librarian 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. students & Librarian. Each one of them would have some exclusive privileges (for e.g. Students can just place the request for a book, but only the librarian has the right to issue the books and also track the return.)

1. Students should be able to
    - Login to the system using his/her credentials
    - Place a request for a particular book
2. The librarian should be able to
    - Login to the system using his/her credentials
    - include new books or remove some books from the inventory
    - manage return and issue book operations

## 2.4 TECHNOLOGY TO BE USED

➢ Front End & Web Components:– 1. Java Classes coding

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

➢ Databases(any one):- 1. Oracle10g

# 3 IMPLEMENTATION IN J2EE LOT

## 3.1 SUMMARY OF THE FUNCTIONALITY TO BE BUILT

The participants need to develop the Online LiMS by building the functionality incrementally in each of the course modules of J2EE 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 | Introduction to Software Engineering | 0.5 | | Analyze the Case study using SDLC phases. |
| 3 | Web Basics (HTML 5,CSS 3, JavaScript, XML) | 4.5 | 1 | |
| 4 | Oracle Basics | 4 | 1 | Creating relevant database tables |
| 5 | OOP & UML | 1.5 | 1 | Creating relevant Use case and class diagrams |
| | Programming Foundation with Pseudo code + Web Basics +Oracle Basics +OOP & UML Test | 1 | | |
| | Core Java 8 & Development Tools (Junit, Log4j) | 10 | 2 | Developing Business components (java classes). Coding for test classes & testing the functionality using JUnit |
| | Core Java 8 + Dev Tools + OOP/UML Test | 1 | | |
| 8 | Servlets | 3.5 | 2 | Project specific implementation not needed 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 J2EE LOT is as follows: Here screen refers to the Console Screen. Convert your HTML Pages into Console Screen using Java classes. (One java class for one HTML page design )

1. Course: HTML, JavaScript
   a. Develop the following screens:

      i. Home page screen: Home page for the LiMS which provides the options for login option.
      ii. Login Screen: Allows the valid user or librarian to logon to the system and display the Main option screen.
      iii. Main option screen: For students, this screen will display options for Books Registration screen only. For librarian, this screen will display options for Books Inventory & Books Transaction screens.

      iv. Books Registration Screen: This screen allows the user/students to place a request for a book.

      v. Books Inventory Screen: This screen allows the librarian to add/delete and modify books in the library system.

      vi. Books Transaction Screen: This screen allows the librarian to issue a book to a user and also mark the return of the book issued. The return date & the fine fields should be disabled as they would be computed automatically.

   b. In this course you need to develop the user interface using console and document the flow of your application including the screenshots 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
   a. Create the following database tables:

      i. Users: This will contain the list of valid users.
      ii. BooksInventory: This contains the list of books available in the library for issue.
      iii. BookRegistration: This will contain the details of the user/student that has placed a request for a book.
      iv. BookTransaction: This will contain the details of issue/return of a book.

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

i. Users: user_id (varchar2(4)), user_name (varchar2(15), password(varchar2(7)), email_id(varchar2(25)), librarian(Boolean)

ii. BooksInventory: book_id(varchar2(4)), book_name(varchar2(20)), author1(varchar2(15),author2(varchar2(15), publisher(varchar2(20)), yearofpublication(varchar2(4))

iii. BooksRegistration: registration_id(varchar2(4)), book_id (varchar2(4)), user_id(varchar2(4)), registrationdate(date)

iv. BooksTransaction: transaction_id(varchar2(4)), registration_id(varchar2(4)), issue_date(date), return_date(date), fine(number(3))

3. Course: OOP & UML

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

4. Course: Core Java 8 + Developer Tools

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. Registration request: This component will allow the user to place a request for a book. It will record the request in the Books Registration table along with registration date.

iii. Issue/Return: This component will allow the librarian to issue a book to a user and also mark the return of the book issued. While issuing the book the return date should be computed automatically as 14 days after issue date. When the book is returned, the fine should be computed as per the policy (Re. 1 per day of delay). This transaction along with the details (return date, fine, etc.) should be recorded in the BooksTransaction table. b. Develop test classes for testing the following functionality i. Login ii. Issue Book. iii. Place a request. c. Test the application using JUnit. d. Configure Logger to log the status of an application

5. Course: Core Java Classes + Developer Workbench

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

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 miniproject 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:<br>• 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** |