

**JEE
Mini Project
Banking System
(BS)**

Document Control

<u>Reference No:</u>			
Security Classification:	Internal		
Issue Date:			
Author(s):	Name	Title	
	Uma Maheswari P	Manager – T & D	
Reviewer(s):	Name	Title	
	Training Team	All Designations in Training	
Issuer(s):	Name	Title	
Issuer Signature(s):			
Distribution:	Name	Title	
	Team	All Designations in Training	
Document History:	Date	Revision	Change
	Nov 7 ,13	0.01D	Initial draft
	June 6 ,15	0.02D	Changes in Miniproject document content as per the upgraded courses
	June 28,16	0.03D	Changes in Miniproject document content as per the upgraded courses

Table of Contents

1	Introduction	4
1.1	Setup Checklist for Mini Project	4
1.2	Instructions	4
2	Problem Statement	5
2.1	Objective	5
2.2	Abstract of the project	5
2.3	Functional components of the project	5
2.4	Technology used	6
3	Implementation in JEE LOT	7
3.1	Summary of the functionality to be built	7
3.2	Guidelines on the functionality to be built	8
3.3	Evaluation and assessment parameters	13

1 INTRODUCTION

This document outlines a mini project for the JEE LOT. The project is to develop a Banking System (OBS). 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 9i client and access to oracle 9i server
- JDK 8
- Eclipse Luna
- JUnit 4.0, MAVEN
- WildFly

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 and documentation provided with WildFly.
- 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 Banking System (OBS)

2.2 ABSTRACT OF THE PROJECT

This project is aimed at developing an Banking System (OBS) for Account holders and Bank admin. **Banking** allows account holders to view mini/detailed statements and do fund transfer .It allows the bank admin to create account and view all transactions on a secure website

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 types of users who would access the system viz. Account Holder and Bank administrator. Each one of them would have some exclusive privileges as follows.

1. Account holders

- Login into the system using his/her credentials.
- View Mini /Detailed statement of all the accounts (Multiple accounts , if any can be viewed)
- Request for change in communication address/mobile number for bank account
- Request for cheque book
- Track service request
- Fund Transfer
- Change password

2. Bank Admin

A set of administrators are assigned for managing the system. An admin has been assigned a set of privileges to manage the system. An admin can perform the following functionalities:

- Create a new Account upon request.
- Can view the reports of overall transactions on daily/monthly/quarterly/yearly basis.

2.4 TECHNOLOGY USED

- *Front End & Web Components:-*
 1. Core Java
- *Business Logic Components and Services :-*
 1. Java Beans
- *Application Servers:-*
 1. WildFly
- *Databases:-*
 1. Oracle 9i

3 IMPLEMENTATION IN JEE LOT

3.1 SUMMARY OF THE FUNCTIONALITY TO BE BUILT

The participants need to develop the **OBS** 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, 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 Test		
6	Core Java 8 & Development Tools(Junit, Log4j)	2	Developing Business components (java classes). Coding for test classes & testing the functionality using JUnit
8	Core Java 8 & Development Tools + OOP/UML Test		
9	Servlets	2	Developing the application using the prototypes. Converting the HTML web pages to jsp pages and java classes (business components) to java beans. Integrating jsp web pages with business components to complete the entire functionality. Building the web applications component using MAVEN build script.
10	JSP		
11	Developer Workbench (PMD, MAVEN)		
12	Servlets + JSP + Dev Workbench Test		

13	Basic Spring 4.0	1	Prepare document for presentation.
14	Basic Spring Test		
15	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. **Core Java (Duration: 10 hours)**

Develop the following screens:

- **Login Screen:**

- i. **Login / Sign In:** If the supplied user credentials are valid, Homepage must be displayed, else appropriate error message must be displayed in the same page. More than 3 attempts must lock the account.

Forgot password option: This should ask the user to answer a specific question which was specified during registration. On answering the security question correctly, a new login password is generated, which is sent to the user via email.(As sending the email is out of scope a dummy password “sbq500#” can be verified in the code The user needs to login using this password and change password for future use. User credential details are assumed to exist in the database

- ii. **Home Page:**

- On successful user authentication the homepage must be displayed according to the type of use. For the Account holder the Homepage would display balance amount with the following links
 - View Mini/Detailed statement
 - Change in address/mobile number
 - Request for cheque book
 - Track service request
 - Fund Transfer
 - Change password

- iii. **View Mini /Detailed statement page**

- Mini Statement:
 - This will show the last 10 transactions
 - Detailed Statement:
 - This will show the transaction details for the period selected

- Account holder can have multiple accounts .He/She should be able to view statements of all his/her accounts
- iv. Request for change in communication address/mobile number for bank account page.
- Displays current communication and Mobile number details and allows the user to change the address of communication and mobile number
- v. Request for cheque book:
- This Accepts the request and displays service request number.
 - A cheque book request is always in Open state. Other valid states are dispatched, issued or returned.
- vi. Track service request page:
- This link helps the user to know the status of the service requested like request for cheque book
 - Following should be displayed at the end of the page
1. Only Service Request numbers pertaining to your Account can be viewed.
 2. To check the status of a particular request, please enter the Service Request number.
 3. To check the status of all Service Requests OR If you do not remember the Service Request number, please select Account Number and click on submit option
 4. Only the latest 20 Service Requests raised in last 180 days will be displayed.
- vii. Fund Transfer page
- This page Provides the following options
 - Your own bank account across India:
 - Allows transferring from user's own accounts.
 - This page displays two drop down boxes each listing out all accounts of that particular user. One is a 'from' dropdown box to select the account from which he/she want to transfer funds. The other is 'pay' dropdown box to select the account to which he/she want to transfer the funds.
 - Other account of same bank across india:

- Funds can be transferred to other accounts of the same bank .On clicking this option "Make a Transfer" page is displayed which has the following steps.

1. Select the payee (person user want to pay to) from the list of payees in the drop down.

In case user did not register for the payee, he/she should click 'add a payee' link provided in the same page.This link allows the user to enter payee account number, account nickname and submit. Payee Registration request gets accepted and remains in "Pending Confirmation" status. If valid payee account number is entered URN(unique Registration Number) is sent to registered mobile(consider dummy URN "abc345"). To confirm a Payee registration, user must enter the URN pin received in the registered mobile(abc345). After confirming Payee Registration it is added to payee list (shows accountId concatenated with nick name).

2. Click the Pay button: user will be prompted to enter the transaction amount and transaction password. On entering the correct transaction password, transaction will be completed(must be validated for sufficient balance)

1. The maximum amount of funds that can be transferred per day is Rs. 10 lakhs.
2. There is no limit on Funds transfer to "your own Bank Account"

viii. Change password

- Password change option allows the user to provide a new password for login.

For the Bank Admin following pages must be provided

ix. Create New Account page:

- This page must accept account holder name,address, mobile number, email-id,account type ,opening balance and generate the account number .

x. View transactions of all accounts

- This link displays the transaction details of all account on daily /monthly/yearly basis

2. Course: Oracle (Duration: 5 hours)

- a. Create the following database tables:
 - i. Account Master: This will contain details of the Account holder and account details
 - ii. Transactions : This will contain details of all transactions (credit and debit)
 - iii. Service_Tracker :This will contain the details of all the services requested, and the status .
- b. The structure of the above listed tables is as follows
 - i. **Account Master** : Account_ID NUMBER(10) Account Type VARCHAR2(25), Account_Balance NUMBER(15) ,Open_Date DATE
 - ii. **Customer** : Account_ID NUMBER(10) , customer_name VARCHAR2(50), Email VARCHAR2(30), Address VARCHAR2(100), Pancard VARCHAR2(15)
 - iii. **Transactions**: Transaction_ID NUMBER ,Tran_description VARCHAR2(100), DateofTransaction DATE , TransactionType VARCHAR2(1) TranAmount NUMBER(15) ,Account_No NUMBER(10)
 - iv. **Service Tracker**: Service_ID NUMBER, Service_Description VARCHAR2(100),Account_ID NUMBER, Service_Raised_Date DATE ,Service_status VARCHAR2(20)
 - v. **User Table**: Account_ID NUMBER ,user_id NUMBER,login password VARCHAR2(15),secret_question VARCHAR2(50),Transaction_password VARCHAR2(15),lock_status VARCHAR2(1)
 - vi. **Fund Transfer**: FundTransfer_ID NUMBER ,Account_ID NUMBER(10) ,Payee_Account_ID NUMBER(10), Date_Of_Transfer DATE, Transfer_Amount NUMBER(15)
 - vii. **PayeeTable**: Account_Id NUMBER ,Payee_Account_Id NUMBER, Nick name VARCHAR2(40)

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 OBS application.

4. Course: Core Java 8 & Development 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.
 - View Mini Statement: This component will show the last 10 transactions
 - Detailed Statement: This component will show the statements between the period of date accepted from the account holder
 - ii. Change in request of the address for communication/Mobile number: This component will display the existing communication address and mobile number. This should allow the user to edit the changes with validation
 - iii. Request for cheque book: This Component will accept the request and displays service request number.
 - iv. Track service request: This component will show the status of the service requested like request for cheque book
 - v. Fund Transfer: This component will allow the user to transfer the amount from his own account or from other bank account across India
 - vi. Change password: This component will allow the user to provide a new password for login.
 - vii. Create New Account page: This component will accept account holder name, address, mobile number, email-id, account type, opening balance and generate the account number.
 - viii. View transactions of all accounts: This component displays the transaction details of all account on daily /monthly/yearly basis
- b. Develop test classes for testing the following functionality
 - i. Login
 - ii. Change communication address
 - iii. View Mini Statement.
- c. Test the application using JUnit.
- d. Configure Logger to log the status of an application

5. 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. 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):

This mini project will be done in groups of five to 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 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