**Bank Management System**

[Objective: 2](#_Toc146967737)

[Abstract: 2](#_Toc146967738)

[Users of Bank Management System 3](#_Toc146967739)

[Functionality available to the Customer 3](#_Toc146967740)

[Modules in the Bank Management System 3](#_Toc146967741)

[User Authentication Module: 3](#_Toc146967742)

[Account Registration Module: 4](#_Toc146967743)

[View Balance Module: 4](#_Toc146967744)

[Deposit Funds Module: 4](#_Toc146967745)

[Withdraw Funds Module: 4](#_Toc146967746)

[Transfer Funds Module: 4](#_Toc146967747)

[NFRs & SLAs: 4](#_Toc146967748)

[Testing: 5](#_Toc146967749)

[TDD/BDD Approach: 5](#_Toc146967750)

[Atomic Design in React 5](#_Toc146967751)

[Other Considerations: 7](#_Toc146967752)

[Screenshots of Bank Management System 7](#_Toc146967753)

[User Authentication: 7](#_Toc146967754)

[CLI Application 7](#_Toc146967755)

[Web Application 8](#_Toc146967756)

[Account Registration Module: 8](#_Toc146967757)

[CLI Application 8](#_Toc146967758)

[Web Application 8](#_Toc146967759)

[Check Balance Feature: 9](#_Toc146967760)

[CLI Application 9](#_Toc146967761)

[Web Application 9](#_Toc146967762)

[View Statement Module: 9](#_Toc146967763)

[CLI Application 9](#_Toc146967764)

[Web Application 10](#_Toc146967765)

[Deposit Funds Module: 11](#_Toc146967766)

[CLI Application 11](#_Toc146967767)

[Web Application 11](#_Toc146967768)

[Withdraw Funds Module: 11](#_Toc146967769)

[CLI Application 11](#_Toc146967770)

[Web Application 12](#_Toc146967771)

[Transfer Funds Module: 12](#_Toc146967772)

[CLI Application 12](#_Toc146967773)

[Web Application 13](#_Toc146967774)

[Logout Feature: 13](#_Toc146967775)

[CLI Application 13](#_Toc146967776)

[Web Application 13](#_Toc146967777)

[Reference: 13](#_Toc146967778)

# Objective:

Develop an end-to-end full-stack web application that is a **Bank Management System** using ReactJS, Spring Boot, Spring REST and Spring Data JPA.

Implement CI-CD with Jenkins.

Dockerize the application.

Deploy Spring Back End and ReactJS Front End applications to AWS.

# Abstract:

Bank Management System Project is an online software application that provides easy, accurate and fast access to the banking system.

This Application will initially provide a Menu-Driven Console Interface to a User. The User can perform functions like Create an Account, Login, View Balance, View Statement, Withdraw, Deposit and Intra Bank Transfer of Funds.

Over the course of five Sprints, various features and functionalities will be introduced like Hibernate ORM replacing JDBC, introduction of REST APIs, transformation to a Spring Framework project, CI/CD with Jenkins, Performance Testing with JMeter till finally it will be deployed as a Full Stack Application with a ReactJS Front End on Amazon S3 and a Spring Boot Back End on an Amazon EC2 instance. Development will follow the TDD approach. NFRs will be emphasized throughout and the system will be expected to generate Unit and BDD-Cucumber test reports in every Sprint.

Before the use of this online banking system, the user should complete Registration. A user holding the bank account is supposed to generate a unique customer Id and secured password for logging in to the software.

The following facilities provided by the Online Bank Management software enhance running banking system and make it more easy, accessible and effective.

* The bank customers can view their account details such as account id, type of account, available balance in the account, etc. from anywhere via internet.
* The software is made to display deposited cash and date of deposition.
* Online Bank Management System is designed to give detailed information on cash withdrawals and deposits.
* Online money transfer is one of the most desired facilities of bank customers. So, this online software provides online money transaction facility.
* Moreover, this project is capable of displaying history of transaction with information such as transaction date, particulars, amount and type.

# Users of Bank Management System

There is one user in the Bank Management System):

* Customer: With Full Access to own account. Can transfer funds to any other customer within the bank.

## Functionality available to the Customer

**These are the features available to a customer:**

* Login/Logout
* Create Account
* View Balance
* View Statement
* Deposit Funds
* Withdraw Funds
* Transfer Funds

# Modules in the Bank Management System

## User Authentication Module:

This module provides all the functionality related to customer login. Customer Id and Password will be validated against the record in the database.

**Features of Login Module:**

* Login
* Logout

## Account Registration Module:

The main purpose for developing this module is to create a new bank account. Customer details input by the prospective customer will be stored in the appropriate database tables after validation.

## View Balance Module:

The main purpose of this module is providing all the functionality related to viewing the account balance.

Transaction records will be retrieved from the database and displayed as formatted output in the console or browser depending on the functionalities described in the Sprint.

## Deposit Funds Module:

The main objective of this module is to manage Deposits. Customer will be able to deposit funds to their own account and the account balance will be updated in the database.

## Withdraw Funds Module:

The main objective of this module is to manage Withdrawals. Customer will be able to withdraw funds from their own account if they have sufficient balance and the account balance will be updated in the database.

## Transfer Funds Module:

Customer can make intra bank transfers based on an account number and the IFSC code.

# NFRs & SLAs:

* The system must support 1-page/5-page requests per second
* The system will need to support 15 user logins per second
* The system will need to support 150 transactions per second.
* Concurrent user session – 3000
* Page view times < 3s
* Availability - 99.999
* Responsive web design - supported view port 767px (Mobile and Desktop)
* CI&CD Pipeline(s) to be deployed via Jenkins | Docker.
* Different customers should not be able to access each other’s data.
* Secure coding standards to be followed.
* User friendly error pages should be displayed to the end users
* Automate Performance Testing with JMeter
* **Only one session per user** at a time or **no concurrent session per user**. If the user tries to open a new session, then either an alert is shown or his previous session is closed.

# Testing:

* Smoke and Sanity Testing
* Unit Testing and Regression
* System Testing
* [JaCoCo](https://www.eclemma.org/jacoco/) - Code coverage > 80%
* Performance Test with JMeter
* Unit Test Reports with JUnit and Spring Boot Testing Framework
* Error and Exception handling
* Test Data preparation, management and data cleanup should be in place

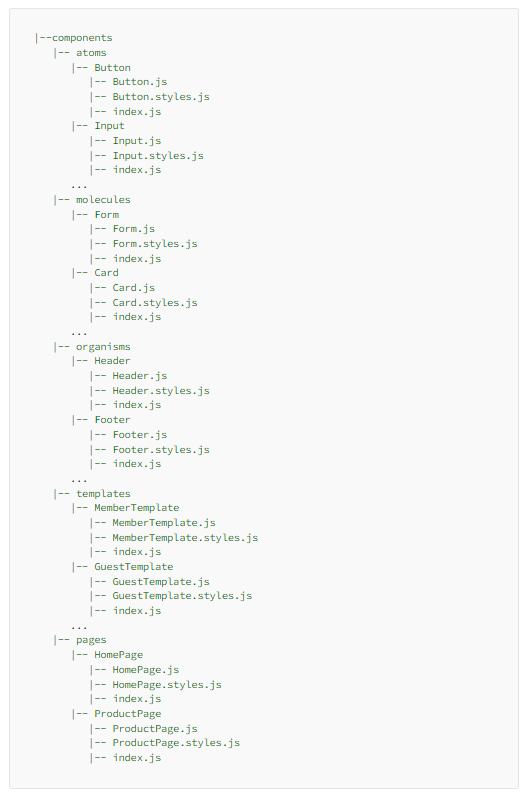
# TDD/BDD Approach:

* TDD with Junit to be followed for Java application development.
* TDD with Jest to be followed for ReactJS application development.
* Behavior Driven Development (BDD) to utilize the Cucumber framework.

# Atomic Design in React

Participants should have a working knowledge of the five distinct levels of atomic design — atoms > molecules > organisms > templates > pages — and how it maps to a React application’s component-based architecture.

This is an example of a project hierarchy that uses the principles of Atomic Design while separating the logic and styles of the components:



Below Hands On Assignments are mandatory:

Week-5 Day-3

|  |
| --- |
| Lab 25 - Build a React Application with Material UI following Atomic Design — Part 1 |
| Lab 25 - Build a React Application with Material UI following Atomic Design — Part 2 |
|  |

# Other Considerations:

* System should support upload of raw data in CSV format.
* Up to 50,000 employee records need to be stored
* The below link allows you to download sample csv files ranging from 100 records to 5000000 records:-

<https://eforexcel.com/wp/downloads-16-sample-csv-files-data-sets-for-testing/>

* Other links that may be helpful for obtaining raw data: -

<https://gist.github.com/kevin336/acbb2271e66c10a5b73aacf82ca82784>

<https://sample-videos.com/download-sample-csv.php>

<https://wsform.com/knowledgebase/sample-csv-files/>

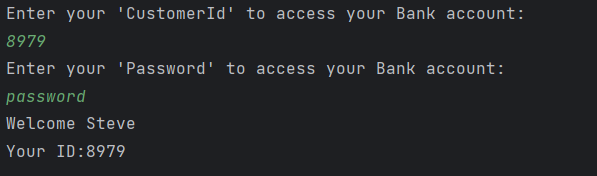
* If no one csv suits your requirement, you can compose your datasets from multiple CSVs.
* You may import only selected columns from a CSV into tables in the database.

# Screenshots of Bank Management System

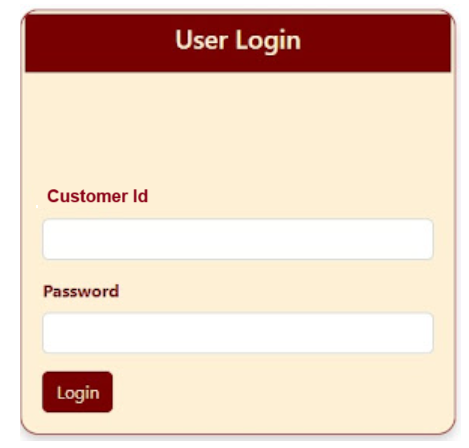
\*\* Participants are free to use their own layout, styling etc. but essential details must be displayed.

## User Authentication:

### CLI Application

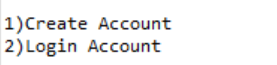


### Web Application



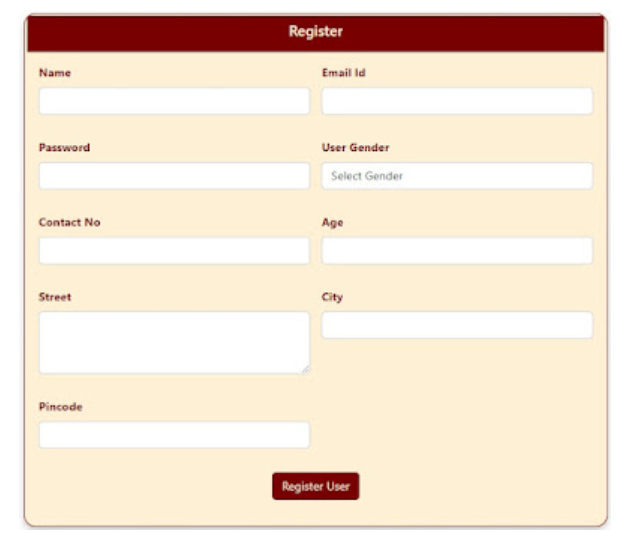
## Account Registration Module:

### CLI Application

****

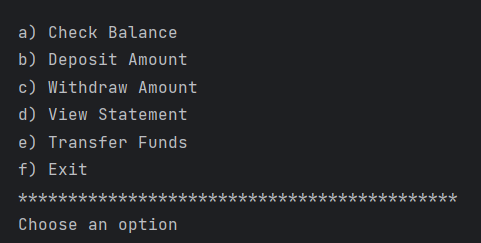
**Same fields to be input as the Web Application.**

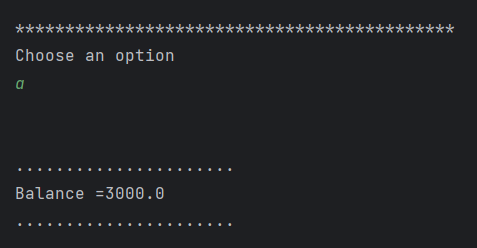
### Web Application

****

## Check Balance Feature:

### CLI Application



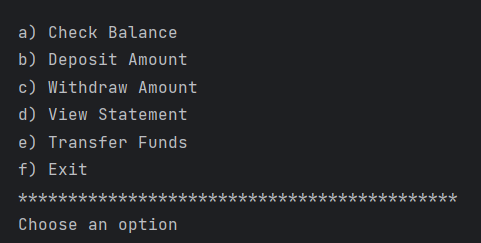
****

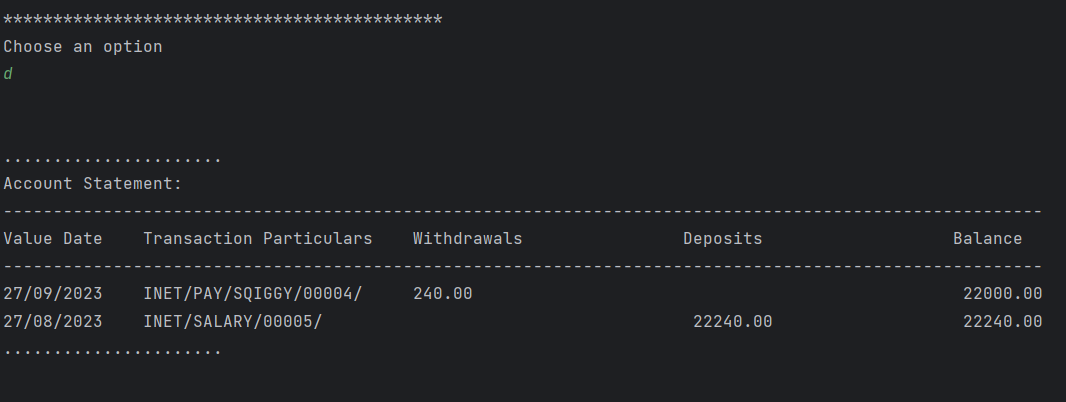
### Web Application

Displayed on the landing page after logging in to the system.

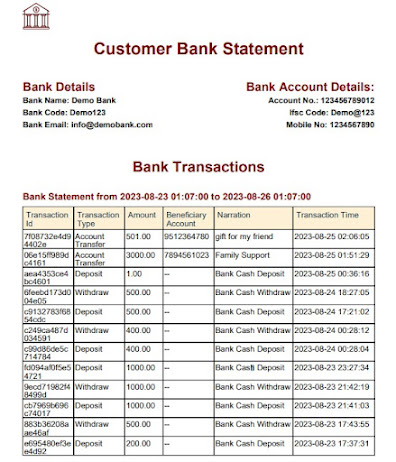
## View Statement Module:

### CLI Application



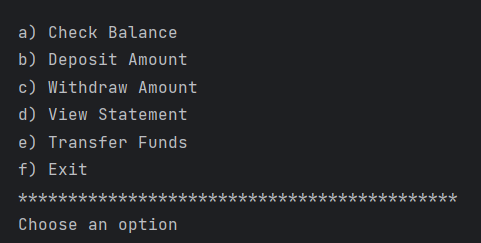
****

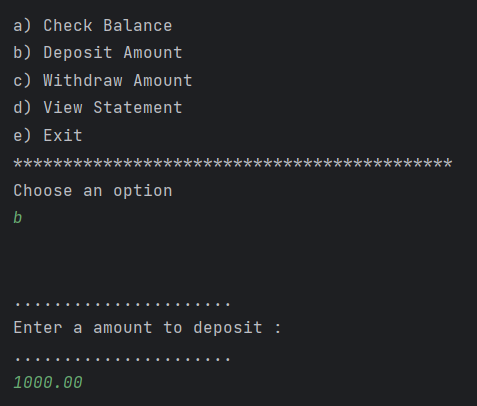
### Web Application



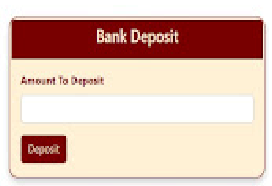
## Deposit Funds Module:

### CLI Application



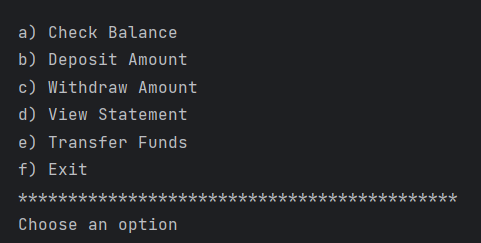
****

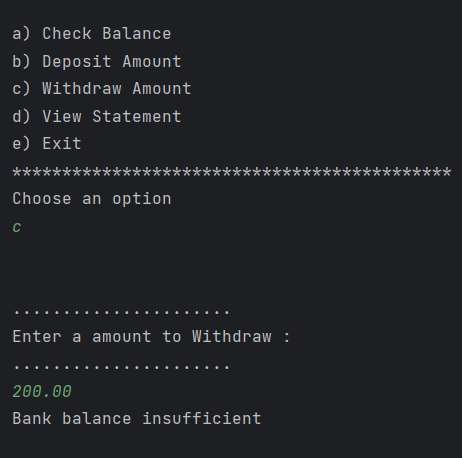
### Web Application

****

## Withdraw Funds Module:

### CLI Application



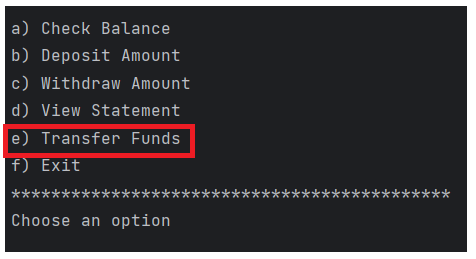
****

### Web Application

****

## Transfer Funds Module:

### CLI Application

****

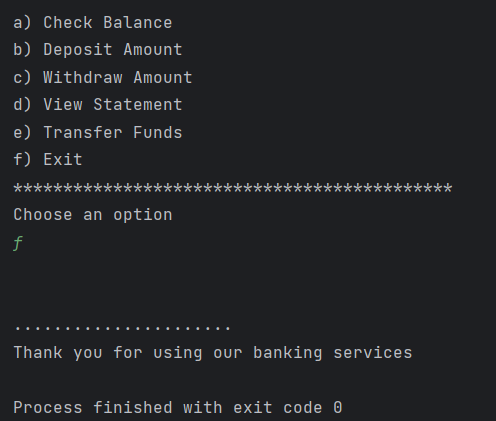
**Same fields to be input as the Web Application.**

### Web Application



## Logout Feature:

### CLI Application



### Web Application

On Logout, customer will be redirected to the Login page.

## Reference:

ASDE\_Training\_USER\_STORIES - 2023.xlsx : – Goals and Tasks for each of five Sprints in the Learning Phase.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*