

# **Advance Java Case Study – Credit Card Management System**

# Advanced Java Case Study – Credit Card Management System

## Table of Contents

### Contents

Welcome!	3
<b>1. Introduction – Architecture Overview</b>	<b>4</b>
<b>2. Architectural Goals</b>	<b>5</b>
<b>3. Architecture Overview</b>	<b>7</b>
Case Study – Credit card System in a bank	9
General Description	9
User Stories	9
<b>User Story 1.1: View Home Page</b>	<b>9</b>
<b>User Story 1.2: Apply for Add on Card</b>	<b>10</b>
<b>User Story 1.3: View card statement</b>	<b>12</b>
<b>User Story 1.4: View payment status</b>	<b>12</b>
<b>User Story 1.5: Cancellation request</b>	<b>14</b>
<b>User Story 1.6: View Card History</b>	<b>15</b>

# Advanced Java Case Study – Credit Card Management System

## Welcome!

Welcome to the Advance Java Case Study exercise for Credit card management System in a bank.

This case study exercise includes six user stories. You need to perform the activities mentioned in the user stories and submit your results.

Before working on the case study exercise, take look at the Architecture Overview for Credit card management System in a bank.

**Lab Set up :- It is expected that you use the set up you have created while going through your learning journey for each of the individual courses.**

## Architecture Overview

### **1. Introduction – Architecture Overview**

#### **1.1 Purpose**

The purpose of the Architecture Overview, as represented by the diagram below, is to:

- Communicate a conceptual understanding of the target IT system
- Provide a high-level shared vision of the architecture and scope of the proposed IT system
- Explore and evaluate alternative architectural options
- Enable early recognition and validation of the implications of the architectural approach
- Facilitate effective communication between different communities of stakeholders and developers
- Facilitate orientation for new people who join the project

#### **1.2 Scope**

This Architecture Overview is documented for Credit card management System in a bank application.

#### **1.3 Intended Audience**

This document is intended for Developers, Architects, Technical Advisory Board, Project Manager, and Designers.

#### **1.4 Overview**

The Architecture Overview diagram represents the governing ideas and candidate building blocks of an IT system and enterprise architecture. It provides an overview of the main conceptual elements and relationships in an architecture, including candidate subsystems, components, nodes, connections, data stores, users, and external systems.

# Advanced Java Case Study – Credit Card Management System

## 2. Architectural Goals

### 2.1 Strategic Drivers

The strategic drivers are as follows:

- Using a layered architecture, which ensures parallel development of multiple layers. Each layer is independent of other layers, which means that we can change the technology of one layer without effecting other layers. For example, we can easily replace JSP in the presentation layer with ReactJS.
- Ensures low coupling between the components. Ensuring high cohesion, which enables that the responsibilities of a given element are strongly related and highly focused.
- Ensure separation of concerns and distribution of responsibilities.

# Advanced Java Case Study – Credit Card Management System

## 2.2 Architectural Drivers

### 2.2.1 Technology Platform

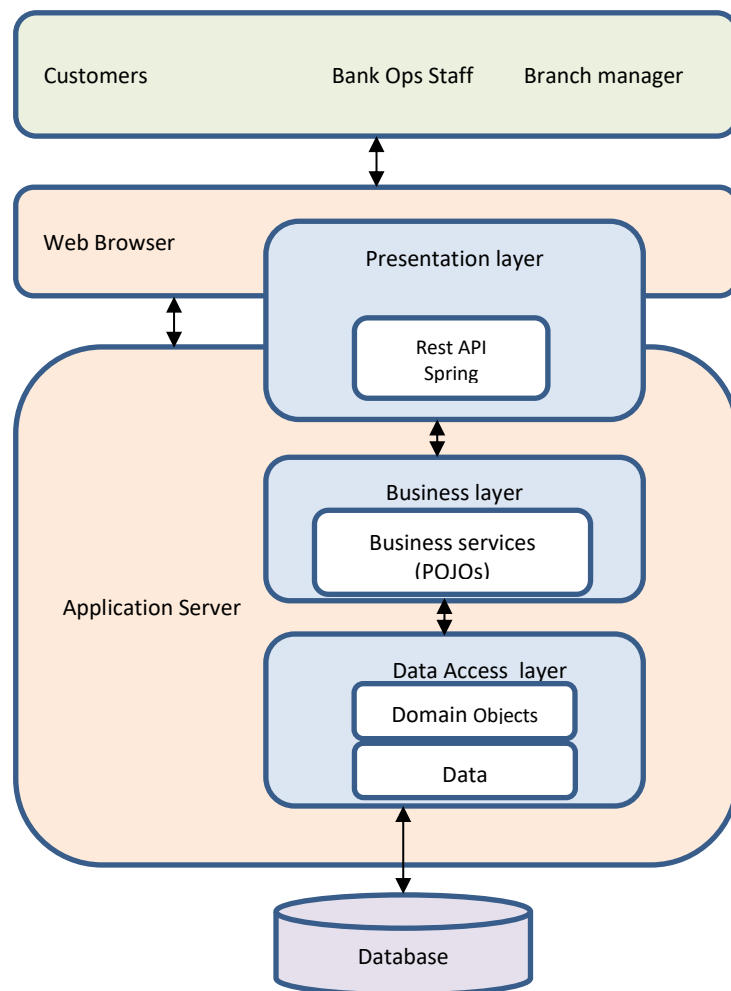
Development platform	A Java-JEE/Stack built around: <ul style="list-style-type: none"><li>• Java Enterprise Edition</li><li>• Spring Framework<ul style="list-style-type: none"><li>• Spring Boot</li><li>• Spring Data MySQL</li></ul></li><li>• Eclipse or STS</li><li>• JUnit</li></ul>
Application server	Apache Tomcat or any application server
Relational database server	MySQL

# Advanced Java Case Study – Credit Card Management System

## 3. Architecture Overview

### 3.1 Application Layers

The diagram given below will provide a clear idea of the different layers of the application.



# Advanced Java Case Study – Credit Card Management System

## **Presentation Layer:**

The presentation layer is responsible for the interaction with outside actors (via the User Interface).

The technology used in the presentation layer is html/jsp coupled with Spring Boot and Spring Rest.

## **Service Layer:**

Service layer consists of business service components which implement the business logic of the application.

## **Data Access Layer:**

This layer is responsible for interaction with the database and performs database-specific operations

**(CRUD).**

## **Persistence Layer:**

This layer is not actually an individual layer. It is clubbed with Data Access layer and it uses Data MySQL to retrieve, save, and update data in the database.



# Advanced Java Case Study – Credit Card Management System

## Case Study Exercise

## Case Study - Credit card management System

### General Description

The Credit card management System is used by a credit card company Manager to track Credit cards. Different types of cards are provided and tracked on parameters, such as transactions per month per Credit card, Credit card limit, total money spent in a month, cash withdrawn if any and payment status of a card holder. Based on this information, the manager wants to categorise the Credit cards holders for business development.

### User Stories

---

#### User Story 1.1: View Home Page

##### *Description*

*As a branch Manager, I want to view the application home page when I log in so that I have easy access to the features of the application and can quickly see relevant information.*

The manager should see the home page of the application after entering the web application's URL.

The home page should contain the following:

1. Name of application

# Advanced Java Case Study – Credit Card Management System

2. Reserved area for functions. Place holders should be placed instead of actual links for functions. Functions will be added through succeeding user stories.
3. Reserved area for default views. Nothing to be displayed for now. Views to be displayed will be determined by succeeding user stories.

## *Out-of-scope*

Actual content of reserved areas is out of scope and will be specified by other stories.

## *Dependencies*

None

## *Acceptance Criteria*

1. Base application URL leads to home page.
  2. Home page displays web application name.
  3. Home page has visible areas for functions and default views.
- 

## **User Story 1.2: New/renew Credit card issue**

### *Description*

*As a bank Manager, I want to add a Credit card so that I can manage Credit card information.*

### *Functional Requirements*

Credit card profile fields should include the following:

Field	Type	Required?
First Name	String	Yes
Middle Name	String	No
Last Name	String	Yes
Birth Date	Date	Yes
Card type	String	Yes
Credit Limit	Double	Yes
Issue date	Date	Yes
Expiry date	Date	Yes

# Advanced Java Case Study – Credit Card Management System

In addition, the system should generate a BRANCH for each active Credit card.

## UI

1. Credit card entry form (KYC) should have its own screen.
2. User should receive feedback that a Credit card/ addon was successfully issued.
3. Credit limit to be divided in case of addon card
4. User should receive feedback if a Credit card was not added for the following reasons:
  - a. Credit card already exists – User should be alerted
  - b. Missing required fields – User should be alerted as to which fields are missing
  - c. Invalid field values – User should be alerted as to why a field value is wrong
5. If Credit card was not added for some other reason, a generic error message should be displayed

## Validation

1. A Credit card cannot be entered into the system twice. Credit card identity is based on first name, middle name, last name, and birth date. Credit cards with all four identical fields are considered to be the same Credit card.
2. Birth date should not be later than current date.

## *Out-of-scope*

1. Legal age validation

## *Dependencies*

None

## *Acceptance Criteria*

1. User can use Credit card to the system as long as it passes validation.
  2. All functional requirements are met.
-

# Advanced Java Case Study – Credit Card Management System

## User Story 1.3: Search Credit card

### *Description*

*As a Branch Manager I want to search for Credit cards in the system so that I can select Credit card to manage.*

### *Functional Requirements*

Search by first name, last name, position or any combination of the three. Search result should match all supplied search criteria (AND).

### UI

1. One field for each search criteria
2. One Search button
3. One Clear button to clear search criteria
4. Executing a search should not clear search criteria
5. Search results should display Credit card UID plus all fields specified in Story 1.2
6. If no matching results are found, “0 results found” should be displayed

### *Out-of-scope*

N/A

### *Dependencies*

Story is dependent on Story 1.2: Add Credit card, unless Credit card entities are already in the database.

### *Acceptance Criteria*

1. User should be able to search for Credit cards
2. All other functional requirements are fulfilled

---

## User Story 1.4: View/Edit Credit card Profile

### *Description*

*As a Branch Manager, I want to view and edit Credit card profiles so that I can change Credit card details when needed.*

# Advanced Java Case Study – Credit Card Management System

## *Functional Requirements*

All Credit card fields except BRANCH should be editable. Validation rules for Credit card fields from Story 1.2 must still be followed  
UI

1. Credit card to be viewed/edited must be selected from search results (see Story 1.3)
2. View/Edit Credit card form must be in its own screen.
3. View/Edit form fields must be pre-populated with Credit card information.
4. All Credit card fields should be visible, including CID.
5. CID should only be displayed. It should not be editable.
6. User should receive feedback if Credit card was successfully edited.
7. User should receive feedback if a Credit card was not updated for the following reasons:
  - a. Identical Credit card found – User should be alerted if editing a Credit card would make it identical to an existing Credit card
  - b. Missing required fields – User should be alerted as to which fields are missing
  - c. Invalid field values – User should be alerted as to why a field value is wrong
8. If Credit card was not edited for some other reason, a generic error message should be displayed.
9. If Credit card was not edited, form fields should not be reset.

## *Out-of-scope*

1. Legal age validation

## *Dependencies*

Story 1.3

## *Acceptance Criteria*

1. User can select a Credit card to view/edit from search results.
  2. User can view/edit a Credit card.
  3. All functional requirements are fulfilled.
-

# Advanced Java Case Study – Credit Card Management System

## User Story 1.5: Add Credit card transaction Details

### Description

*As a Branch Manager, I want to add interest compounded yearly, added monthly for a Credit card so that I can track a Credit card's details in the system.*

It is possible for a Credit card to have add ons and each add on will be combined with the main card for generation of monthly statement. The due date of payment should be calculated and displayed with the due amount. A penalty will be imposed for delayed payment. These should be recorded in the system. Credit card details include the card type (entry level, international, national and privilege etc.), statistics, and description should be available.

### Functional Requirements

#### Assumptions

1. It will be up to the branch Manager to define the list of Credit cards for categorizing them .

#### Fields

Field	Type	Required?
Card name	String	<i>See below</i>
Type	Numeric	<i>See below</i>
Description	String	<i>See below</i>
No of holders	long	

#### Types of information

1. Credit card details – This is basic info. Only one entry per Credit card per account can be added. It is possible for the value to be zero) or upto 11. Description is optional.
2. Transaction details – A Credit card can spend/withdraw any number of times in a month. Description is optional.
3. Any account that has not been operated for last 6 months.
4. Any Other info – This covers anything not covered above. More than one entry is allowed. Values can be zero also. Description is required.

# Advanced Java Case Study – Credit Card Management System

## Validation

1. See section above for validation rules for the different accounts/Credit cards.

## *Out-of-scope*

1. Transaction amount
2. Calculating total interest earned in a financial year

## *Dependencies*

### Story 1.3

## *Acceptance Criteria*

1. User can add details as per functional requirements
- 

## **User Story 1.6: View Interest History**

### *Description*

*As a Branch Manager, I want to view the interest history of a Credit card so that I can generate a report on how much an Credit card has earned for the company.*

Only total statistics per category will be displayed for chosen Credit card.

### *Functional Requirements*

#### UI Flow

1. User searches for Credit card
2. User enters type id
3. New screen shows total statistics per month for each year
4. User can choose to return to Home Page or Step 4

#### UI

1. Total as well as average earning per card types should be shown. No breakdown needed.
2. If no details for a account can be found, that account is not displayed. However, in the case where details have zero value, that account should still be displayed.

# Advanced Java Case Study – Credit Card Management System

## Validation

1. When entering date range, user cannot enter an end date that occurs before start date

## *Out-of-scope*

Breakdown of account wise details