

Project: Bank Loan Management System

1. Overview

The **Bank Loan Management System** is a web-based application that streamlines the process of loan application, approval, disbursement, and repayment tracking. The system is designed for banking institutions to handle various types of loans such as personal loans, home loans, and vehicle loans.

This application adopts the **MVC architecture** and is compatible with **Java (Spring MVC)** and **.NET (ASP.NET Core MVC)** frameworks, ensuring flexibility for implementation.

2. Core Modules Description

The system is divided into five key modules, each catering to a distinct aspect of loan management.

2.1 Customer Management

This module focuses on customer account creation, KYC (Know Your Customer) verification, and account maintenance. It acts as the foundation for associating customers with loan applications.

2.2 Loan Product Management

This module is designed for managing loan products offered by the bank, such as personal loans, home loans, and vehicle loans. It includes details like interest rates, tenures, and eligible amounts.

2.3 Loan Application Management

This module facilitates customers to apply for loans. It manages loan application workflows, from submission to approval or rejection, and tracks associated details like loan products and requested amounts.

2.4 Repayment Management

The repayment management module automates the creation of repayment schedules, records payments, and calculates outstanding balances. It ensures timely tracking and compliance with repayment terms.

2.5 Reporting and Auditing

This module generates insightful reports for administrators, including loan statistics, repayment statuses, and customer portfolios, ensuring transparency and aiding decision-making.

3. Module-Level Design

3.1 Customer Management Module

Purpose: Handles customer registration, account details, and KYC verification.

- **Controller:**

- CustomerController
 - registerCustomer()
 - getCustomerDetails()
 - updateCustomerProfile()
- **Service:**
 - CustomerService
 - Processes KYC details and manages customer accounts.
- **Model:**
 - **Customer Entity**
 - Attributes:
 - customerId (PK)
 - name
 - email
 - phone
 - address
 - kycStatus (PENDING, VERIFIED)

3.2 Loan Product Management Module

Purpose: Allows administrators to create and manage loan product details.

- **Controller:**
 - LoanProductController
 - addLoanProduct()
 - updateLoanProduct()
 - getLoanProductDetails()
- **Service:**
 - LoanProductService
 - Implements logic for managing loan types and terms.
- **Model:**
 - **LoanProduct Entity**
 - Attributes:
 - loanProductId (PK)
 - productName
 - interestRate
 - minAmount
 - maxAmount
 - tenure (in months)

3.3 Loan Application Module

Purpose: Facilitates loan application and approval workflows.

- **Controller:**
 - LoanApplicationController
 - applyForLoan()

- `getApplicationStatus()`
 - `processLoanApplication()`
- **Service:**
 - `LoanApplicationService`
 - Validates application details and handles approval workflows.
- **Model:**
 - **LoanApplication Entity**
 - Attributes:
 - `applicationId` (PK)
 - `customerId` (FK)
 - `loanProductId` (FK)
 - `loanAmount`
 - `applicationDate`
 - `approvalStatus` (PENDING, APPROVED, REJECTED)

3.4 Repayment Management Module

Purpose: Tracks loan repayment schedules and outstanding balances.

- **Controller:**
 - `RepaymentController`
 - `getRepaymentSchedule()`
 - `makePayment()`
 - `getOutstandingBalance()`
- **Service:**
 - `RepaymentService`
 - Generates repayment schedules and updates balances.
- **Model:**
 - **Repayment Entity**
 - Attributes:
 - `repaymentId` (PK)
 - `applicationId` (FK)
 - `dueDate`
 - `amountDue`
 - `paymentDate`
 - `paymentStatus` (PENDING, COMPLETED)

3.5 Reporting Module

Purpose: Provides financial reports for customers and administrators.

- **Controller:**
 - `ReportController`
 - `generateLoanReport()`
 - `generateRepaymentReport()`
 - `generateOutstandingLoansReport()`
- **Service:**

- ReportService
 - Aggregates and processes data for reporting.
- **Model:**
 - Generates data summaries from other modules.

4. Database Schema

4.1 Table Definitions

1. Customer Table

```
CREATE TABLE Customer (
  customerId INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  email VARCHAR(100),
  phone VARCHAR(15),
  address TEXT,
  kycStatus ENUM('PENDING', 'VERIFIED')
);
```

2. LoanProduct Table

```
CREATE TABLE LoanProduct (
  loanProductId INT AUTO_INCREMENT PRIMARY KEY,
  productName VARCHAR(50),
  interestRate DECIMAL(5, 2),
  minAmount DECIMAL(10, 2),
  maxAmount DECIMAL(10, 2),
  tenure INT
);
```

3. LoanApplication Table

```
CREATE TABLE LoanApplication (
  applicationId INT AUTO_INCREMENT PRIMARY KEY,
  customerId INT,
  loanProductId INT,
  loanAmount DECIMAL(10, 2),
  applicationDate DATE,
  approvalStatus ENUM('PENDING', 'APPROVED', 'REJECTED'),
  FOREIGN KEY (customerId) REFERENCES Customer(customerId),
  FOREIGN KEY (loanProductId) REFERENCES LoanProduct(loanProductId)
);
```

4. Repayment Table

```
CREATE TABLE Repayment (
  repaymentId INT AUTO_INCREMENT PRIMARY KEY,
  applicationId INT,
  dueDate DATE,
  amountDue DECIMAL(10, 2),
```

```
paymentDate DATE,  
paymentStatus ENUM('PENDING', 'COMPLETED'),  
FOREIGN KEY (applicationId) REFERENCES LoanApplication(applicationId)  
);
```

5. Local Deployment Details

1. Environment Setup:

- Install MySQL or SQL Server for database management.
- Set up Java (JDK) or .NET SDK (ASP.NET Core).
- Configure database credentials in the application configuration file.

2. Steps to Deploy Locally:

- Clone the repository.
- Use Maven (Java) or Visual Studio (ASP.NET Core) to build the application.
- Start the application server (Tomcat or Kestrel).
- Access the application at <http://localhost:8080> (Java) or <http://localhost:5000> (.NET).

6. Conclusion

The **Bank Loan Management System** is a robust solution for streamlining loan operations in the BFS domain. Its modular design ensures easy scalability and adaptability for different banking needs, providing a user-friendly experience for both customers and administrators.