1. Overview

The Loan Origination and Management System (LOMS) is a web-based application designed for banks

and financial institutions to automate the loan lifecycle. It encompasses functionalities such as loan

application, credit evaluation, loan disbursal, repayment tracking, and overdue management.

This system is built using the MVC architecture and is compatible with both Java (Spring MVC) and .NET

(ASP.NET Core MVC) frameworks.

2. Core Modules Description

The system consists of five core modules to manage different stages of the loan lifecycle.

2.1 Customer Management

This module manages customer information and tracks loan applicants.

2.2 Loan Application Management

Handles loan applications, including validation and document upload.

2.3 Credit Evaluation

Assesses applicant creditworthiness using predefined parameters and generates credit scores.

2.4 Loan Disbursal

Manages the approval and disbursal of loans to customers.

2.5 Loan Repayment and Overdue Management

Tracks loan repayment schedules and handles overdue accounts.

3. Module-Level Design

3.1 Customer Management Module

Purpose: Manages customer profiles and loan applicant details.

• Controller:

o CustomerController

▪ registerCustomer()

▪ updateCustomerDetails()

▪ getCustomerDetails()

• Service:

o CustomerService

▪ Validates customer data and interacts with the database.

• Model:

o Customer Entity

▪ Attributes:

▪ customerId (PK)

▪ name
email

▪ phone

▪ address

3.2 Loan Application Management Module

Purpose: Handles loan applications and associated document uploads.

• Controller:

o LoanApplicationController

▪ submitApplication()

▪ getApplicationStatus()

▪ uploadDocuments()

• Service:

o LoanApplicationService

▪ Processes loan applications and ensures data integrity.

• Model:

o LoanApplication Entity

▪ Attributes:

▪ applicationId (PK)

▪ customerId (FK)

▪ loanType (Personal, Home, Auto)

▪ amountRequested

▪ status (Pending, Approved, Rejected)

3.3 Credit Evaluation Module

Purpose: Evaluates the creditworthiness of loan applicants.

• Controller:

o CreditEvaluationController

▪ evaluateCreditScore()

▪ getCreditReport()

• Service:

o CreditEvaluationService

▪ Implements algorithms for credit score calculation.

• Model:

o CreditScore Entity

▪ Attributes:

▪ scoreId (PK)

▪ customerId (FK)

▪ creditScore

▪ evaluationDate

3.4 Loan Disbursal Module

Purpose: Manages loan approvals and fund disbursal.

• Controller:

o LoanDisbursalController

▪ approveLoan()

▪ disburseLoan()
getDisbursalDetails()

• Service:

o LoanDisbursalService

▪ Handles loan approval workflows and disbursal logic.

• Model:

o LoanDisbursal Entity

▪ Attributes:

▪ disbursalId (PK)

▪ applicationId (FK)

▪ disbursalAmount

▪ disbursalDate

▪ approvedBy

3.5 Loan Repayment and Overdue Management Module

Purpose: Tracks repayment schedules and overdue accounts.

• Controller:

o RepaymentController

▪ recordRepayment()

▪ getRepaymentHistory()

▪ handleOverdueAccounts()

• Service:

o RepaymentService

▪ Manages repayment schedules and overdue penalties.

• Model:

o Repayment Entity

▪ Attributes:

▪ repaymentId (PK)

▪ loanId (FK)

▪ amountPaid

▪ paymentDate

▪ dueDate

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
);
```

2. LoanApplication Table

```
CREATE TABLE LoanApplication (
 applicationId INT AUTO_INCREMENT PRIMARY KEY,
 customerId INT,
 loanType ENUM('Personal', 'Home', 'Auto'),
 amountRequested DECIMAL(10, 2),
 status ENUM('Pending', 'Approved', 'Rejected'),
 FOREIGN KEY (customerId) REFERENCES Customer(customerId)
);
```

3. CreditScore Table

```
CREATE TABLE CreditScore (
 scoreId INT AUTO_INCREMENT PRIMARY KEY,
 customerId INT,
 creditScore INT,
 evaluationDate DATE,
 FOREIGN KEY (customerId) REFERENCES Customer(customerId)
);
```

4. LoanDisbursal Table

```
CREATE TABLE LoanDisbursal (
 disbursalId INT AUTO_INCREMENT PRIMARY KEY,
 applicationId INT,
 disbursalAmount DECIMAL(10, 2),
 disbursalDate DATE,
 approvedBy VARCHAR(100),
 FOREIGN KEY (applicationId) REFERENCES LoanApplication(applicationId)
);
```

5. Repayment Table

```
CREATE TABLE Repayment (
 repaymentId INT AUTO_INCREMENT PRIMARY KEY,
 loanId INT,
 amountPaid DECIMAL(10, 2),
 paymentDate DATE,
 dueDate DATE,
 FOREIGN KEY (loanId) REFERENCES LoanDisbursal(disbursalId)
);
```

5. Local Deployment Details

1. Environment Setup:

o Install MySQL or SQL Server for database management.

o Install Java (JDK) or .NET SDK (ASP.NET Core).

o Configure database connections in application.properties (Java) or appsettings.json (.NET).

2. Deployment Steps:

o Clone the repository.

o Build the project using Maven (Java) or Visual Studio (ASP.NET Core).

o Run the application server.

o Access the application at http://localhost:8080 (Java) or http://localhost:5000 (.NET).

6. Conclusion

The Loan Origination and Management System streamlines the end-to-end loan process, from application

to repayment. Its modular design ensures scalability and ease of maintenance, providing a robust solution

for financial institutions