# Project Proposal

# On

# Bank Management System

**Guided By:-**

Anuj Kumar

**Created By:-**

Tushar Sharma , Yashark Vikal , Uttkarsh Mavi

**AF_id :-** AF04991839 , AF04990815 , AF04991809

**Batch code :-** ANP-D2405

**Course code :-** ITPR

# Table of Content

1. Title of theProject.....

2. Introduction.....

3. Objective

4. Project Category

5. Analysis

➢ Modules and Description

➢ Database Design

➢ ER Diagram

➢ Data Flow Diagram

6. Complete Structure

➢ Process Logical Diagram

7. Platform Used

➢ Hardware Requirement

➢ Software Requirement

8. Future Scope

9. Bibliography

## 1. Title of the Project

Bank Management System

## 2. Introduction

A **Bank Management System** is a software application designed to handle the daily activities of a bank in a faster and more organized way. In traditional banking, tasks like creating customer accounts, managing deposits and withdrawals, checking balances, and processing loans require a lot of manual work. This often leads to delays, errors, and difficulty in maintaining records.

This system helps the bank store all customer and account information in one place, so staff can easily access, update, and manage data. It makes banking operations quicker, reduces paperwork, improves accuracy, and increases security. With the help of this system, banks can provide better and more reliable services to their customers.

The project will be developed using :-

**Core Java**

**JDBC** for database connectivity

**MySQL** for data storage

**Maven** as the build and project management tool.

## 3. Objective

- To manage all banking data in one place.
- To reduce manual work and save time.
- To improve accuracy of customer and account data.
- To make data search and retrieval faster.
- To increase security of banking information.
- To provide quick and better customer service.
- To maintain clear records of all transactions.
- To connect all banking modules smoothly.
- To build a system that can grow in the future.

## 4. Project Category

This project falls under the category of **Application Software (Database Management System)**.

It is a Console-Based Banking System built using the following technologies : -

● Core Java for logic implementation

● JDBC (Java Database Connectivity) for connecting Java with MySQL

● MySQL Database for storing and managing all banking data

● Maven for project management and dependency handling

## 5. Analysis

## 5.1 Modules and Description

### 1. Customer Management Module

- Customer registration
- Customer profile update
- KYC upload & verification
- Customer status (active/inactive)
- Customer search

### 2. Account Management Module

- Open account (Savings/Current/FD)
- Close account
- View account details
- Change account type
- Balance check

### 3. Transaction Management Module

- Deposit
- Withdraw
- Fund Transfer
- Mini statement
- Full transaction history

### 4. Employee/Staff Management Module

- Add employee
- Update employee details
- Assign roles (Admin / Manager / Staff)
- Employee login credentials
- Salary & designation

**5. Branch Management Module**

- Add new branch
- Assign IFSC/Branch code
- Staff list per branch
- Customer list per branch

**6. Loan Management Module**

- Loan apply
- Loan eligibility check
- Loan approval/rejection
- EMI calculation
- Loan repayment status

**7. Security & Authentication Module**

- Login system (customer + employee)
- Password hashing
- Role-based access control
- Session management
- Activity logs (who logged in, when)
- Security alerts (simulated)

### 5.2 Database Design (Table Format)

1. **customers**

| Columns | Type | Key/Constraints |
|---|---|---|
| customer_id | INT AUTO | PK |
| first_name | VARCHAR(50) | Not Null |
| dob | DATE | Not Null |
| gender | VARCHAR(10) | Not Null |
| phone | VARCHAR(15) | UNIQUE |
| email | VARCHAR(100) | UNIQUE |
| address | TEXT | Not Null |
| kyc_status | ENUM('Pending','Verified','Rejected') | Not Null |
| created_at | TIMESTAMP | Not Null |
| status | ENUM('Active','Inactive') | Not Null |

2. accounts

| Columns | Type | Key/Constraints |
|---|---|---|
| account_id | INT AUTO | PK |
| customer_id | INT | FK |
| branch_id | INT | FK |
| account_type | ENUM('Savings','Current','FD') | Not Null |
| balance | DECIMAL(15,2) | Not Null |
| opened_on | DATE | Not Null |
| status | ENUM('Active','Closed') | Not Null |

3. transactions

| Columns | Type | Key/Constraints |
|---|---|---|
| transaction_id | INT AUTO | PK |
| account_id | INT | FK |
| txn_type | ENUM('Deposit','Withdraw','Transfer') | Not Null |
| amount | DECIMAL(15,2) | Not Null |
| txn_date | TIMESTAMP | Not Null |
| reference_account | INT | FK |
| balance_after_txn | DECIMAL(15,2) | Not Null |

4. employees

| Columns | Type | Key/Constraints |
|---|---|---|

| | | |
|---|---|---|
| employee_id | INT AUTO | FK |
| branch_id | INT | Not Null |
| first_name | VARCHAR(50) | Not Null |
| email | VARCHAR(50) | UNIQUE |
| phone | VARCHAR(100) | Not Null |
| role | ENUM('Admin','Manager','Staff') | Not Null |
| salary | DECIMAL(10,2) | Not Null |
| date_joined | DATE | Not Null |
| status | ENUM('Active','Inactive') | Not Null |

## 5. branches

| Columns | Type | Key/Constraints |
|---|---|---|
| branch_id | INT AUTO | PK |
| branch_name | VARCHAR(100) | Not Null |
| ifsc_code | VARCHAR(20) | UNIQUE |
| address | TEXT | Not Null |
| phone | VARCHAR((15) | Not Null |

## 6. loans

| Columns | Type | Key/Constraints |
|---|---|---|
| loan_id | INT AUTO | PK |
| customer_id | INT | FK |
| loan_type | ENUM('Home','Personal','Vehicle') | Not Null |
| amount | DECIMAL(15,2) | Not Null |
| interest_rate | DECIMAL(15,2) | Not Null |
| tenure_months | INT | Not Null |
| emi_amount | DECIMAL(10,2) | Not Null |
| loan_status | ENUM('Pending','Approved','Rejected','Active','Completed') | Not Null |
| applied_on | DATE | Not Null |
| approved_on | DATE(nullable) | Not Null |

7. loan_payments

| Columns | Type | Key/Constraints |
|---|---|---|
| payment_id | INT AUTO | PK |
| loan_id | INT | FK |
| amount_paid | DECIMAL(10,2) | Not Null |
| payment_date | DATE | Not Null |
| remaining_amount | DECIMAL(15,2) | Not Null |

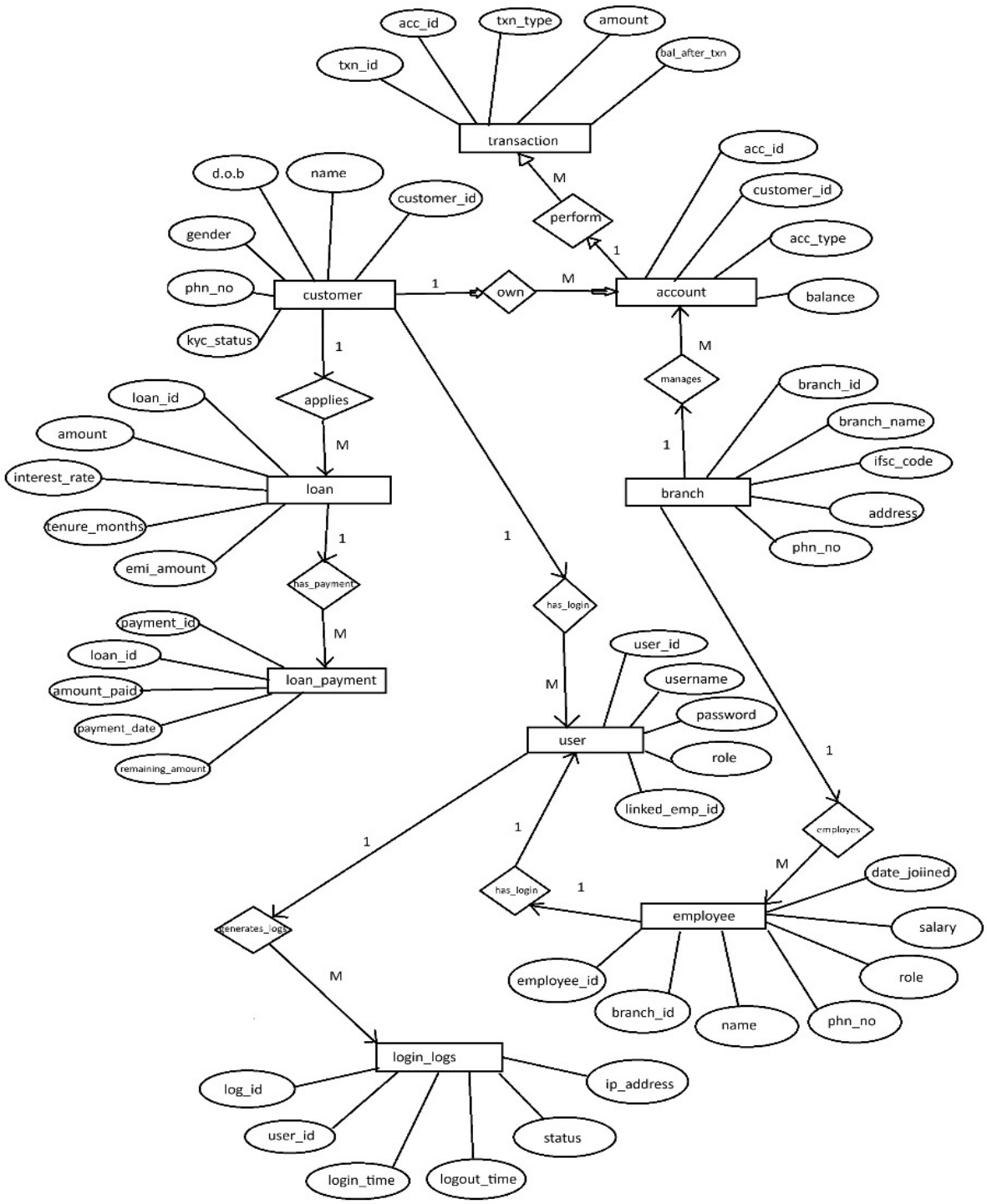
8. users

| Columns | Type | Key/Constraints |
|---|---|---|
| user_id | INT AUTO | PK |
| username | VARCHAR(50) | UNIQUE |
| password_hash | VARCHAR(255) | Not Null |
| role | ENUM('Customer','Employee','Admin') | Not Null |
| linked_customer_id | INT | FK |
| linked_employee_id | INT | FK |
| last_login | TIMESTAMP | Not Null |

9. login_logs

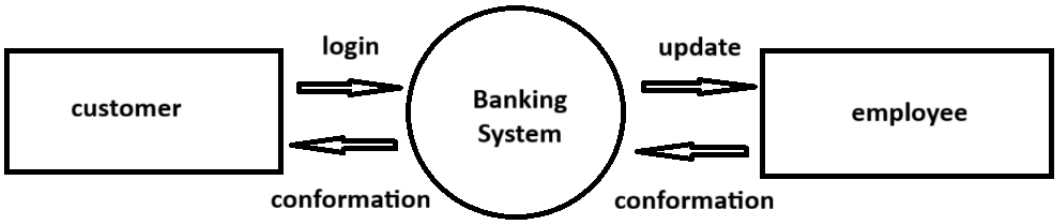
| Columns | Type | Key/Constraints |
|---|---|---|
| log_id | INT AUTO | PK |
| user_id | INT | FK |
| login_time | TIMESTAMP | Not Null |
| logout_time | TIMESTAMP | Not Null |
| status | ENUM('Success','Failed') | Not Null |
| ip_address | VARCHAR(50) | Not Null |

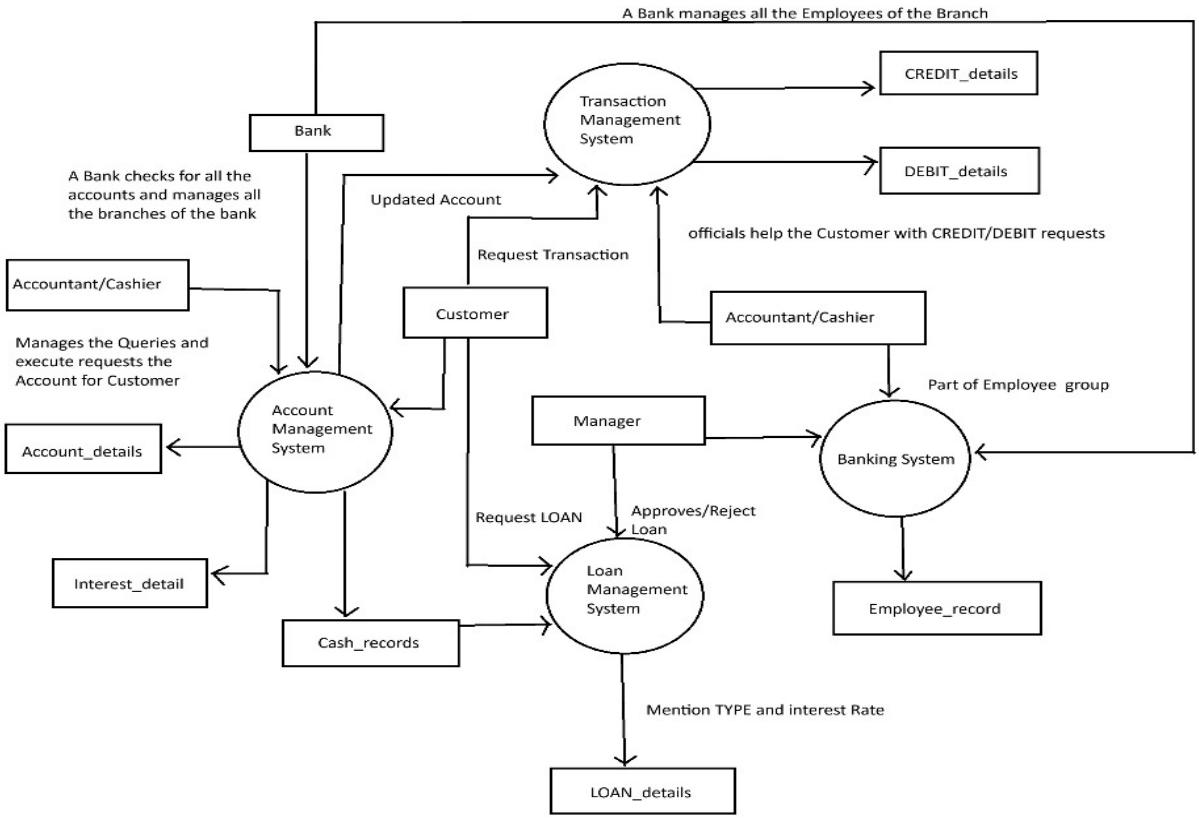
### 5.3 ER Diagram


acc_id
txn_type
amount
txn_id
bal_after_txn
transaction
M
perform
1
acc_id
customer_id
acc_type
d.o.b
name
customer_id
gender
phn_no
customer
1
own
M
account
balance
kyc_status
1
M
manages
branch_id
loan_id
applies
branch_name
amount
M
1
ifsc_code
interest_rate
loan
branch
address
tenure_months
phn_no
1
1
emi_amount
has_payment
has_login
payment_id
M
user_id
loan_id
loan_payment
M
username
amount_paid
password
payment_date
user
role
remaining_amount
linked_emp_id
1
1
employes
1
M
has_login
1
date_joiined
generates_logs
employee
salary
M
employee_id
role
branch_id
name
phn_no
login_logs
log_id
ip_address
user_id
status
login_time
logout_time

## 5.4 Data Flow Diagram (DFD)

### Level 0 DFD




A Bank manages all the Employees of the Branch
Bank
Transaction Management System
CREDIT_details
DEBIT_details
A Bank checks for all the accounts and manages all the branches of the bank
Updated Account
Request Transaction
officials help the Customer with CREDIT/DEBIT requests
Accountant/Cashier
Customer
Accountant/Cashier
Manages the Queries and execute requests the Account for Customer
Part of Employee group
Account Management System
Manager
Account_details
Banking System
Request LOAN
Approves/Reject Loan
Interest_detail
Loan Management System
Employee_record
Cash_records
Mention TYPE and interest Rate
LOAN_details

## 6. Complete Structure

Login

↓

Dashboard

↓

Customer Management → Account Management → Transaction Management

↓

Loan & EMI →  User & Authentication

↓

Branch Management → Employee Management → KYC Verification

↓

Logout

## 7. Platform Used

**Hardware Requirements**

- Processor: Intel i5
- RAM: 16GB
- Storage: 1GB free

**Software Requirements**

- Windows
- JDK
- MySQL Server
- Maven

- IDE :- Eclipse

## 8. Future Scope

- The system can be expanded to support online banking.
- Mobile app versions can be added for easy customer access.
- Advanced security features like OTP and biometrics can be included.
- AI-based fraud detection can be added in the future.
- The system can connect with UPI and digital payment platforms.
- Automated loan approval using credit score analysis can be added.
- Real-time notification services (SMS/Email) can be included.
- More branches of the bank can be connected under one system.
- Cloud storage can be used for faster access and backup.
- New modules like insurance, cards, or investments can be added later.

## 9. Bibliography


- MySQL Documentation
- JDBC API Documentation
- Java Official Documentation
- GeeksforGeeks
- Maven Project Documentation