CS 1103

Database System Project Proposal: Banking Management System

Team members

Riya Bhardwaj

Sanjida Shakhayet

1. Introduction

Throughout this semester, we have learned the essential concepts of database systems, including database schema design, data manipulation, and database connectivity. The objective of this final project is to apply these concepts by creating a **Banking Management System**. This project will involve designing a secure and efficient database schema, inserting and updating records, and implementing a Java-based application using JDBC to interact with the database.

2. Aim of the Project

The primary goal of this project is to design and implement a **Banking Management System** that supports various banking operations. Key features of the system include:

- **Account Management**: Supports the creation of customer accounts, modification of account details, and retrieval of account balances.
- **Transaction Management**: Tracks deposits, withdrawals, and transfers, providing real-time transaction processing.
- **Financial Reports**: Generates reports such as account summaries, transaction histories, and monthly statements.
- **Security**: Implements encryption and secure login features for customer authentication.
- **Interest and Loan Management**: Manages the calculation of interest on savings accounts and processes loan applications.

3. System Requirements

- **Features of the System:**
    - **User Registration & Authentication:** Customers should be able to sign up and log in securely.
    - **Account Management:** Create, view, update, and delete customer accounts.
    - **Transactions:** Support for deposits, withdrawals, and transferring funds between accounts.
    - **Account Balance Inquiry:** Customers should be able to check their balances.

4. Database Design

- **Schema Design** Define the tables for your banking system such as Customer, Account, Transaction, etc.

- Ensure relationships between tables, for example:
    - A customer can have one or more accounts.
    - A transaction is associated with a particular account.

- **Tables Example:**
    - **Customer** (customer_id, name, email, password)
    - **Account** (account_id, customer_id, account_type, balance)
    - **Transaction** (transaction_id, account_id, type, amount, date)

5. Technology Stack

- **Back-End:** using Java for back-end development.
- **Front-End (if applicable):** HTML for a web-based UI.
- **Database:** using SQLite for database.

6. GitHub Repository link

https://github.com/Sanjida-49/Banking-Management-System.git

7. Conclusion

This project will allow us to combine theoretical knowledge of database systems with practical applications in the banking sector. We anticipate challenges such as ensuring data security, handling concurrent transactions, and optimizing system performance. These challenges will provide valuable learning opportunities and improve our problem-solving skills in a real-world context. Ultimately, this project will help us develop a deeper understanding of banking operations and database management, preparing us for future careers in software development and data management.