Higher Institute of Digital Engineering

Data Structures and Algorithms
Dr. Rihab BOUSLAMA

Banking Management System

Project Report

Students :

Yassine Hannachi

Oumayma Hmida

Rabii Zarrouki

Wajdi Bedoui

December 2025

Table of Content :

Introduction :

In this project we developed a banking-management software system — Banking Management System — to simulate core banking operations: customer management, account handling, transactions, loans, and daily transaction logging. Given that manually managing bank data (accounts, loans, branches, employees) is tedious and error-prone, the system offers a computerized approach that ensures consistency and ease of use.

The program is intended as a console-based banking system (in C++) to allow creation/management of customers and employees, facilitate deposits/withdrawals, manage loans, and record daily transactions.

2. Objectives :

1. Allow **account operations**: deposit, withdrawal.
2. Implement **loan management**: loan requests , loan approvals/denials, storing loan data.
3. Manage **employee records** .
4. Gather and analyze **statistics** related to employee accounts and customer accounts.

3. System Features :

The proposed system will have the following features :

- **Customer Management**: add new customers, modify customer data, delete closed accounts.
- **Account Operations**: deposit money, withdraw money, submit loan request.
- **Loan Handling**: queue for loan requests (FIFO), process loan requests (approve / deny), assign loans to customers.
- **Employee Management:** Add, delete, or modify an employee and sort employees.
- **Daily Transactions Log:** maintain a log of all transactions of the day and then finalize at end-of-day.
- **Real-Time Date Access:** retrieve the current system date in order to timestamp transactions and loan requests.
- **Statistics:** gather and analyze key information about both customer accounts and employee records.

4. System Architecture & Design :

The Banking Management System is designed using a modular architecture. This approach keeps the `main()` function simple and focused on coordinating program flow rather than implementing detailed logic.

This modular design improves maintainability, readability, and facilitates future extension, such as adding new features or modules

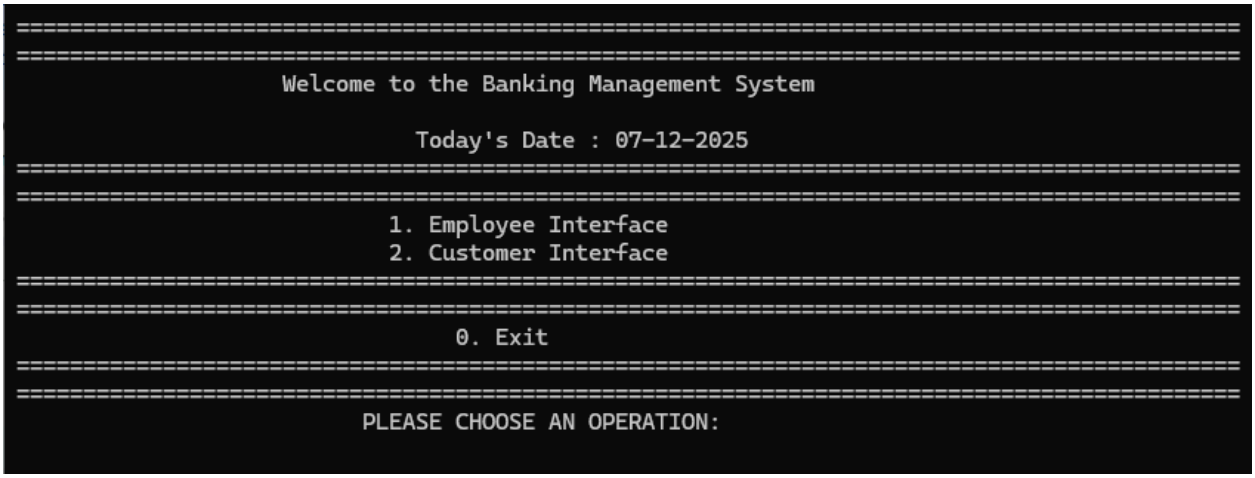
5.Data Structure:

The system uses a variety of data structures to meet different functional requirements:

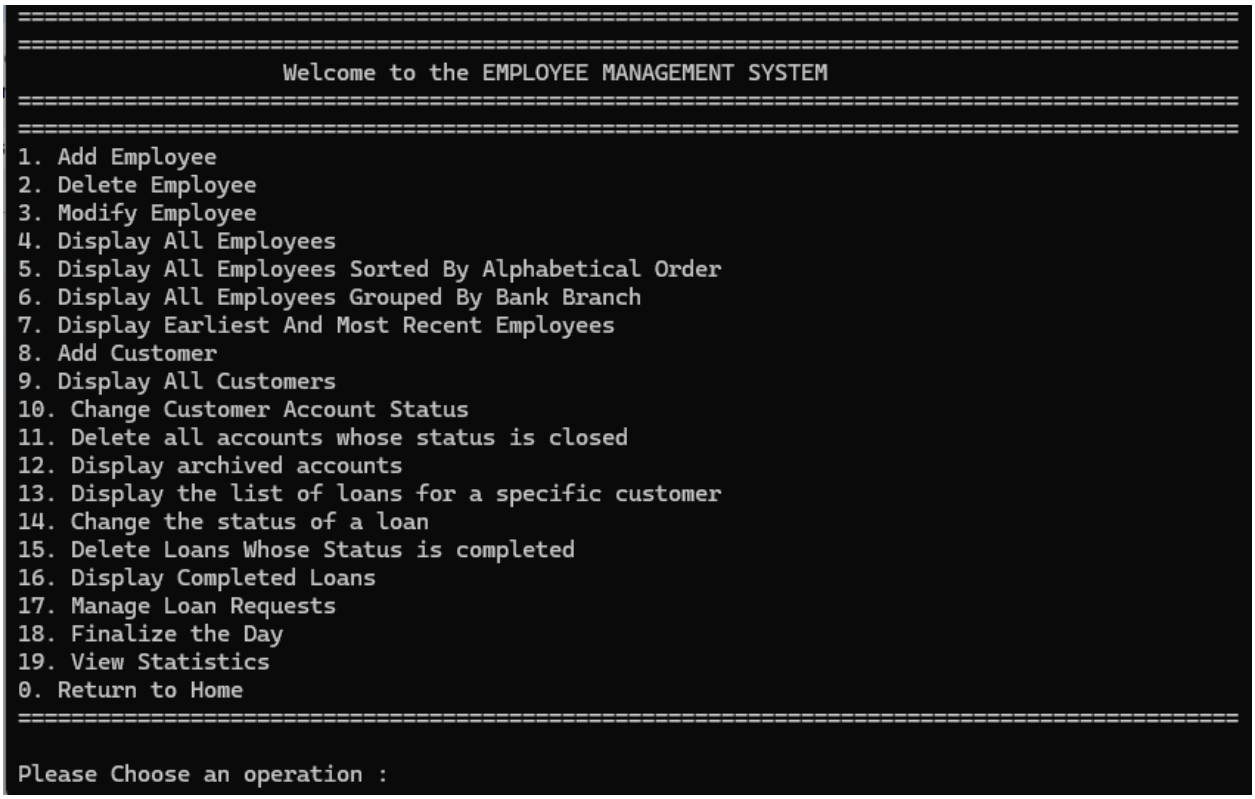
- Structures :
  Customer
  Loan
  Transaction
- Linear Data Structures:
  Array
  Singly Linked List
  Doubly Linked List
  Stack
  Queue
- Auxiliary Modules:
  Data handling
  Date management
  Statistics

6. User Interface Samples:

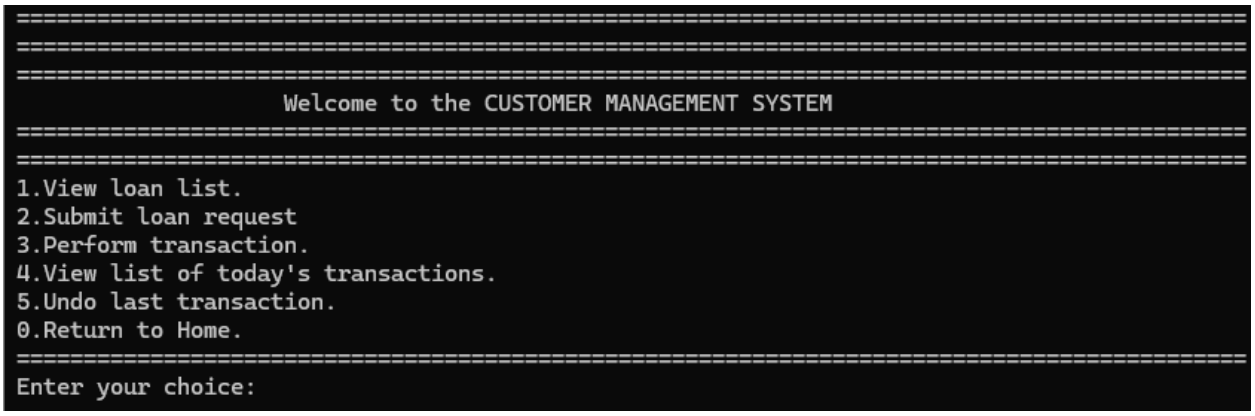
Home Menu :



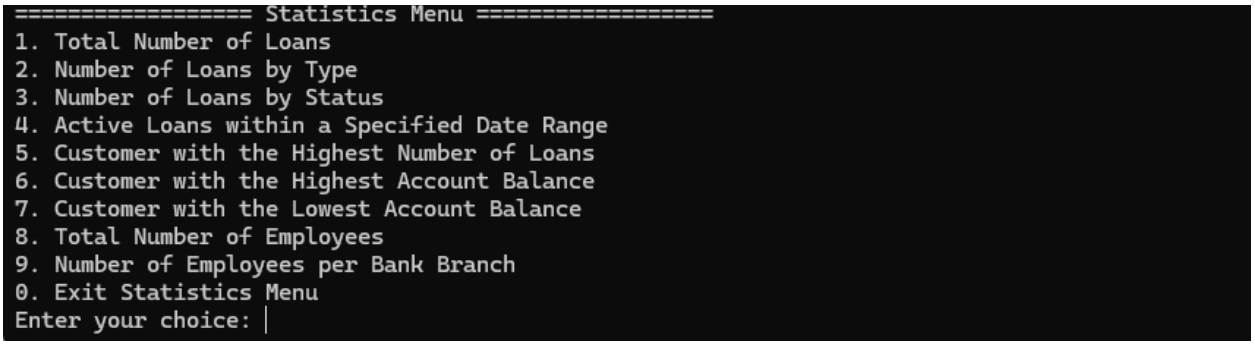
Employee Interface Menu :



Customer Interface Menu once Login Successful :



Statistics Menu :



7.Conclusion:

In this project, our team successfully developed a Banking Management System that simulates core banking operations, including customer account management, loan handling, employee management, daily transaction logging, statistics reporting, and real-time date tracking.

The system demonstrates a modular design with separate, well-structured modules for each functionality, ensuring readability, maintainability, and ease of future expansion.

Throughout the project, we applied good software engineering practices, including modular programming and use of appropriate data structures (queues, stacks, linked lists).

Additionally, our team used GitHub for version control and collaboration, which allowed us to track changes efficiently, work concurrently, and manage the project effectively.

Overall, the project not only met the functional requirements but also provided valuable experience in teamwork, version control, and structured software development.

Possible future improvements include adding a graphical user interface, database integration, enhanced security features, and more advanced statistical reporting.

Github Repository link :
https://github.com/oumaymahmida06-gif/Banking-Management-System-