INTERNSHIP PROJECT-9 DOCUMENT

ON “Bank Management System (JAVA)”

Submitted by:

Navpreet Singh (INTERN)

Submitted To:-

Kanduri Abhinay (FOUNDER)

RITHIN VERMA (CTO)

INDEX

INTRODUCTION:-

The Bank Management System is a console-based application developed in Java that simulates basic banking operations. It enables users to create bank accounts, deposit money, withdraw funds, and view account statements with transaction histories. The project demonstrates the use of object-oriented programming concepts such as classes, objects, encapsulation, and collections. It offers a simple, real-world scenario to practice account management through software.

SOFTWARE REQUIREMENTS:-

- **Programming Language:** Java (version 8 or above)
- **Development Environment:** Any Java IDE (e.g., IntelliJ IDEA, Eclipse, NetBeans) or command-line tools (javac, java)
- **Operating System:** Platform-independent (Windows, Linux, MacOS)
- **Java Runtime Environment (JRE):** Installed (Java SE)
- No external libraries or frameworks are required.

DESIGN:-

The system is designed using object-oriented principles with the following main components:

- **Transaction Class:** Represents individual transactions (deposits or withdrawals) with amount and date.
- **Account Class:** Represents bank accounts with properties such as account number, holder name, balance, and a list of transactions.
- **Bank Class**: Manages multiple accounts using a HashMap for efficient lookup, providing methods to perform operations like adding accounts, deposits, withdrawals, and generating account statements.
- **BankManagementSystem Class:** The main driver class that provides a console-based interactive menu for users to interact with the system.
- The design separates concerns clearly, encapsulates data, and uses collections to manage multiple accounts dynamically.

INPUT:-

Inputs are taken interactively from the console by the user:

- Account details when creating a new account:

1. Account number (String)
2. Holder's name (String)
3. Initial balance (double)

- For transactions:

1. Account number to identify the account
2. Deposit or withdrawal amount (double)

Users choose operation options via numerical menu input.

IMPLEMENTATION:-

The project is implemented using Java with the following functionality:

- Creation of accounts storing initial data.
- Deposit method to add money to accounts, updating balances and transaction logs.
- Withdrawal method allowing users to remove money, checking for sufficient balance.
- Transaction logging with timestamp for each operation.
- Account statements printing current balance and transaction history.
- Use of HashMap<String, Account> to store and quickly access accounts by their number.

Console menu loop to allow interactive use with options for all functions.

TESTING:-

- Manual testing was conducted through console interaction.
- Test cases included:
- Creating multiple accounts with valid and duplicate account numbers
- Depositing various amounts, including boundary cases like zero or negative values.
- Withdrawing amounts within and exceeding current balance to test validations.

- Viewing transactions and balances after multiple operations.
- Tests confirmed correct balance updates and transaction logs.
- Error messages displayed appropriately for invalid operations like insufficient balance or unknown account numbers.

ADVANTAGES:-

- Simple and intuitive console interface for users.
- Demonstrates core OOP concepts with practical applicability.
- Uses collections for efficient data storage and retrieval.
- Easily extendable to add features like authentication, multiple account types, interest calculations.
- Platform-independent due to Java.
- Provides a foundation for understanding real-world banking software design in a simplified manner.

CONCLUSION:-

The Bank Management System project successfully models essential banking operations in Java using object-oriented programming and collections. It provides a functional tool for managing accounts and transactions via an interactive console, reinforcing programming and software design skills. The system is modular, scalable, and serves as a solid base for more complex banking applications.

REFERENCES:-

Oracle Java Documentation: https://docs.oracle.com/en/java/

Java Tutorials by Oracle: https://docs.oracle.com/javase/tutorial/

GeeksforGeeksJavaOOP Articles: https://www.geeksforgeeks.org/java/

JAVA Framework: https://www.tutorialspoint.com/java/java_collections.htm