Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year: 2024), B.Sc. in CSE (Eve)

LAB PROJECT PROPOSAL

Course Title: Object Oriented Programming Lab

Course Code: CSE-202 Section: 223_E1

Student Details

Name		ID
1.	WALIULLAH	223015026

Project Proposal Date : 23-05-2024

Submission Date : 26-05-2024

Course Teacher's Name : Abdullah Al Farhad

[For Teachers use only: Don't Write Anything inside this box]

Project Proposal Status

Marks: …… **Signature: .....**

Comments: ..... **Date: .....**

1. **TITLE OF THE PROJECT PROPOSAL**

   Developing a Bank Management System using OOP in JAVA Programming.

   The Java Bank Management System is a comprehensive software solution designed to streamline the operations of a bank. Built using object-oriented programming principles, this system offers a user-friendly interface for both customers and bank employees to manage various banking activities efficiently.

2. **PROBLEM DOMAIN & MOTIVATIONS**

- # Problem:

- **Account Management:** The system needs to manage various types of accounts such as savings, checking, and fixed deposit accounts. This involves creating new accounts, updating account details, and closing accounts.
- **Transaction Handling:** It should facilitate various transactions like deposit, withdrawal, fund transfer, and viewing transaction history. These transactions need to be securely handled to ensure data integrity and security.
- **Customer Management:** The system should manage customer information including personal details, contact information, and account ownership. It also involves functionalities like adding new customers, updating customer details, and deleting customer records.
- **Security:** Implement security measures such as authentication and authorization to ensure that only authorized users can access the system and perform operations.

- # Motivations:

- **Efficiency:** Automating bank operations reduces manual effort, minimizes errors, and speeds up processes. OOP allows for modular design, making it easier to maintain and extend the system.
- **Accuracy:** By using standardized algorithms and data structures, the system can perform calculations and transactions accurately, ensuring reliability in financial operations.
- **Customer Satisfaction:** Providing a user-friendly interface and efficient services enhance the overall customer experience. Quick access to account information and hassle-free transactions lead to higher satisfaction levels.
- **Data Security:** Implementing secure coding practices and encryption techniques ensures the confidentiality and integrity of sensitive financial data, instilling trust in customers and regulatory authorities.

3. **OBJECTIVES/AIMS**

- **Create a User Interface:** Develop a user-friendly interface for customers to interact with the bank system, enabling them to perform various transactions like deposit, withdrawal, balance inquiry, etc.

- **Implement Account Management:** Design classes for different types of bank accounts such as savings, current, and fixed deposit accounts. Implement functionalities for account creation, deletion, and modification.
- **Handle Transactions:** Incorporate methods to carry out transactions like depositing money, withdrawing funds, transferring between accounts, and viewing transaction history.
- **Ensure Security:** Implement security features such as authentication mechanisms username and password to ensure that only authorized users can access their accounts.
- **Manage Customer Information:** Develop mechanisms to store and retrieve customer information including their personal details, account information, and transaction history securely.
- **Support Multiple Users:** Enable multiple users to access the system simultaneously without data interference or loss.
- **Handle Exceptions:** Implement exception handling to deal with errors and unexpected scenarios gracefully, ensuring smooth functioning of the system.
- **Enable Admin Functions:** Incorporate administrative functionalities such as adding or removing employees, monitoring system logs, and managing system configurations.

4. TOOLS & TECHNOLOGIES

- **Java:** Core programming language.
- **IDE (Integrated Development Environment):** Visual Studio Code (Vs Code)
- **Version Control:** Git with GitHub.
- **GUI (Graphical User Interface) Library:** AWT or Swing for creating the user interface.

5. CONCLUSION

This is a basic structure for a Bank Management System in Java. You can expand upon this by adding features such as different types of accounts, user authentication, database integration for persistent data storage, etc. Additionally, error handling and validation can be improved for real-world applications.