Theory Assignment

Only for course Teacher						
		Needs Improvement	Developing	Sufficient	Above Averag e	Total Mark
Allocate mark & Percentage		25%	50%	75%	100%	5
Clarity	1					
Content Quality	2					
Spelling & Grammar	1					
Organization and Formatting	1					
Total obtained mark						
Comments						

Semester: Spring ……24….. / Fall ………

Student Name: Md. Sourov Hasan

Student ID: 0242220005341153

Batch: 39th **Section: C**

Course Code: SE-221

Course Name: Object Oriented Design

Course Teacher Name: Mr. Akash Ghosh

Designation: Lecturer **Submission Date:21/05/24**

Project Idea

1.Customer Registration and Authentication:

Customers can register with a unique ID, name, and password.
Registered customers can log in using their ID and password.

2.Account Management:

After logging in, customers can:
View their accounts.
Open new accounts.
Add balance to an account.
Withdraw money from an account.

3.Transaction Management:

Each account records transactions such as deposits, withdrawals, and balance additions.
Here's a detailed breakdown of the classes and methods:

Classes and Their Responsibilities

1.Customer:

Represents a bank customer with an ID, name, password, and a list of accounts.
Methods:
authenticate(String password): Validates the customer's password.
addAccount(Account account): Adds a new account to the customer.
getAccounts(): Retrieves the customer's accounts.

2.Account:

Represents a bank account with an account number, balance, and a list of transactions.
Methods:
deposit(double amount): Deposits money into the account.

withdraw(double amount): Withdraws money from the account if the balance is sufficient.
addBalance(double amount): Adds money to the account.
getTransactions(): Retrieves the account's transactions.

3.Transaction:

Represents a transaction with a type (e.g., deposit, withdrawal), amount, and timestamp.
Methods:
toString(): Returns a string representation of the transaction.

4.Bank:

Manages the customers and their accounts.
Methods:
registerCustomer(String id, String name, String password): Registers a new customer if the ID is unique.
authenticateCustomer(String id, String password): Authenticates a customer using their ID and password.
addAccountToCustomer(String customerId, String accountNumber): Adds a new account to a customer.

5.Main:

Contains the main method and handles the user interface through a command-line interface.
Methods:
main(String[] args): Entry point of the program.
register(): Handles customer registration.
login(): Handles customer login.
customerMenu(Customer customer): Displays the menu for logged-in customers.
viewAccounts(Customer customer): Displays the customer's accounts.
openAccount(Customer customer): Allows the customer to open a new account.
addBalance(Customer customer): Allows the customer to add balance to an account.
withdrawMoney(Customer customer): Allows the customer to withdraw money from an account.

Potential Project Ideas

1.Enhanced Banking System:

Add more account types like savings, checking, and fixed deposit accounts.
Implement interest calculation for savings accounts.
Allow money transfers between accounts.

2.Graphical User Interface (GUI):

Create a graphical user interface using JavaFX or Swing to improve user experience.
Add features like charts to display transaction history.

3.Security Enhancements:

Implement stronger password policies and encryption for stored passwords.
Add multi-factor authentication (MFA) for logging in.

4.Notification System:

Implement an email or SMS notification system for transactions and account activities.

5.Loan and Credit Features:

Add features for customers to apply for loans or credit.
Implement a system to calculate loan eligibility and manage repayments.

6.Integration with External APIs:

Integrate with external financial APIs to fetch real-time exchange rates, stock prices, or other financial data.

7.Admin Panel:

Create an admin panel for bank administrators to manage customers and accounts. Include features like generating reports, monitoring transactions, and flagging suspicious activities.

By enhancing and expanding the current system, you can create a comprehensive project that demonstrates a wide range of software development skills.