**Simplified Banking System Project**

**High-Level Description:**

This basic banking system simulation project involves fundamental object-oriented programming (OOP) concepts like classes, objects, methods, variables, inheritance, constructors, and basic control structures. The focus is on simulating simple banking operations like account creation, balance inquiry, PIN validation, and deposits/withdrawals. The project avoids using collections and focuses solely on essential OOP principles.

**Classes Overview:**

* **Account:** Represents a bank account with properties like account number, balance, and PIN.
* **Customer:** Represents a customer who owns an account.
* **ATM:** Simulates an ATM machine that interacts with the Account class to perform banking operations.
* **Bank:** Manages the overall bank operations, such as storing and validating customer information.

**Class Customer class Account**

name: String accountNumber: int

account: Acc. balance: double

getAccount(); pin: int

**class ATM** deposit();

atmId: int withdraw();

atmLocation: String validatePIN();

checkBalance(); **class Bank**

validatePIN(); customers: customer[]

deposit(); addCustomer();

withdraw(); getCustomer();

* **Account Management:** The Account class handles all the operations related to a bank account like deposits, withdrawals, and PIN validation.
* **Customer Management:** The Customer class associates a customer with their respective bank account.
* **ATM Operations:** The ATM class simulates an ATM machine's interaction with an account, providing services like balance inquiry, deposit, and withdrawal.
* **Bank Operations:** The Bank class manages customers and their accounts, providing methods to add customers and retrieve customer information based on their account number.

**Low-Level Design (LLD)**

**1. Class: Customer**

**Attributes:**

* name: String - The name of the customer.
* account: Account - The bank account associated with the customer.

**Methods:**

* Customer(String name, Account account): Constructor to initialize the customer with a name and an account.
* getAccount(): Account: Returns the customer's account.
* getName(): String: Returns the customer's name.

**2. Class: Account**

**Attributes:**

* accountNumber: int - The unique account number.
* balance: double - The current balance in the account.
* pin: int - The PIN associated with the account for security.

**Methods:**

* Account(int accountNumber, double balance, int pin): Constructor to initialize the account with an account number, balance, and PIN.
* getAccountNumber(): int: Returns the account number.
* getBalance(): double: Returns the current balance.
* deposit(double amount): Adds the specified amount to the balance.
* withdraw(double amount): Deducts the specified amount from the balance if sufficient funds are available.
* validatePIN(int enteredPIN): boolean: Validates the entered PIN against the stored PIN.

**3. Class: ATM**

**Attributes:**

* atmId: int - The unique ID of the ATM.
* atmLocation: String - The location of the ATM.

**Methods:**

* ATM(int atmId, String atmLocation): Constructor to initialize the ATM with an ID and location.
* checkBalance(Account account): Prints the current balance of the account.
* deposit(Account account, double amount): Deposits the specified amount into the account.
* withdraw(Account account, double amount): Withdraws the specified amount from the account if sufficient funds are available.
* validatePIN(Account account, int enteredPIN): boolean: Validates the entered PIN against the account's PIN.

**4. Class: Bank**

**Attributes:**

* customers: Customer[] - An array of customers in the bank.
* numOfCustomers: int - The number of customers currently stored.

**Methods:**

* Bank(int size): Constructor to initialize the bank with a specified size (maximum number of customers).
* addCustomer(Customer customer): Adds a new customer to the bank.
* getCustomer(int accountNumber): Customer: Retrieves a customer based on the account number.

**5. Class: Main**

**Attributes:**

* None.

**Methods:**

* main(String[] args): The entry point of the application where the bank, customers, and ATM are initialized and basic operations are performed.