Details of semester project

OBJECT ORIENTED PROGRAMMING

*NAME: MUHAMMAD HAMZA AZEEM*

*ROLL # SP24-BSE-033*

*NAME: MUHAMMAD ARABI*

*ROLL# SP24-BSE-031*

# Necessary Classes in the Software

1. ATM
2. ATMGUI
3. Bank Database
4. User
5. Account
6. Transaction *(abstract class)*
7. Deposit *(inherits from Transaction)*
8. Withdraw *(inherits from Transaction)*

# 

# Specifications of Classes.

**1. ATM**

* **Attributes:**
  + bankDatabase: BankDatabase
  + currentUser: User
  + gui: ATMGUI
* **Methods:**
  + login(userId: int, pin: int): boolean
  + performTransaction(option: int): void
  + start(): void

**2. ATMGUI**

* **Attributes:**
  + atm: ATM
  + panel: List<JPanel>
* **Methods:**
  + showLoginScreen(): void
  + showMainMenu(): void
  + showWithdrawPanel(): void
  + showDepositPanel(): void
  + displayMessage(msg: String): void

**3. BankDatabase**

* **Attributes:**
  + users: List<User>
  + accounts: List<Account>
* **Methods:**
  + authenticateUser(userId: int, pin: int): User
  + getAccount(userId: int): Account
  + updateBalance(userId: int, amount: double): void

**4. User**

* **Attributes:**
  + userId: int
  + name: String
  + pin: int
  + account: Account
* **Methods:**
  + validatePin(pin: int): boolean
  + getAccount(): Account
  + getName(): String

**5. Account**

* **Attributes:**
  + accountNumber: int
  + balance: double
  + transactions: List<Transaction>
* **Methods:**
  + getBalance(): double
  + deposit(amount: double): void
  + withdraw(amount: double): boolean
  + addTransaction(type: String, amount: double): void

**6. Transaction *(Abstract)***

* **Attributes:**
  + date: LocalDateTime
  + amount: double
* **Methods:**
  + getDetails(): String

**7. Deposit *(Inherits Transaction)***

* **Constructor:**
  + Deposit(double amount)
* **Methods:**
  + execute(account: Account): void

**8. Withdraw *(Inherits Transaction)***

* **Constructor:**
  + Withdraw(double amount)
* **Methods:**
  + execute(account: Account): void

# Relationships between different classes

| **Class A** | **Relationship Type** | **Class B** | **Description** |
| --- | --- | --- | --- |
| ATM | Association | BankDatabase | ATM uses the database for user/account verification |
| ATM | Association | User | ATM keeps track of the currently logged-in user |
| ATM | Association | ATMGUI | ATM interacts with GUI for user operations |
| BankDatabase | Aggregation | User, Account | Holds user and account lists |
| User | Association | Account | Each user has an associated account |
| Account | Aggregation | Transaction | Account maintains a list of transactions |
| Transaction | Inheritance | Deposit, Withdraw | Subclasses implement specific transaction logic |

# UML Diagram

