

# ATM

NOTE: I assumed my ATM's Bank name to be ABC Bank.

## ➤ CORE TAKEAWAYS FROM ATM:

### a) **USER's Perspective:**

1. ATM users should be able to log in with their account number and password.
2. ATM users should be able to view account balance, withdraw cash, and deposit funds.
3. ATM users should be able to edit their basic profile.

### b) **ATM's Perspective:**

1. ATM should be able to fetch users' details and store them securely.
2. ATM should be able to hold notes of several denominations.
3. ATM should be able to make and update changes after each transaction whether it's withdrawal, deposition or profile update.
4. ATM should be able to keep track of all transactions.
5. ATM should be able to accept coins too for deposit.

## ➤ WORKING OF ATM:

1. Firstly, ATM asks whether it's an admin or a customer. If it's **admin** then the admin interface opens up.

Admin has choices to

- a) Make ATM unserviceable
  - b) Open the Cash Tray
  - c) Continue to the Customer Interface
2. If it's a **customer** then the jumps to the customer interface.
  3. Customer has to input his/her **account number** and **password**.

If the Customer inputs an incorrect account number, then he can re-enter but only with 4 tries left. If he couldn't be able to input the correct account number, then he's asked to visit the nearest ABC Bank and rectify his/her issue with the account number.

4. If the Customer is able to enter the correct account number, then he should enter the corresponding password. He's given **5 tries** to enter his/her correct password otherwise his account gets **BLOCKED** and he/she has to contact customer care for support.
5. If the customer is not using his/her bank account for several days then his/her account is automatically set to **HIBERNATE** mode. He/she then has to contact customer care to set his/her account to live.
6. The Customer has options to choose from:
  - a) **View Account Balance**
  - b) **Withdraw Cash**
  - c) **Deposit Money**
  - d) **Update Basic Profile**
  - e) **Get a Statement for his/her transactions**
7. **Account Balance** option shows how much amount is left in his/her account.
8. **Withdraw Cash option** allows customers to withdraw an amount less than the maximum amount set by ABC Bank.
9. **Deposit Money option** allows customers to deposit money ranging from coins of one rupee, or two rupees to notes of thousands.
10. **Update Basic Profile option** allows customers to update their basic credentials like Name, Phone Number, Address etc.
11. **Get a Statement for his/her transactions option** allows customer to get a statement for all the transactions witnessed by customer.
12. Customer can go back to menu to perform multiple operations in a single visit.
13. Customer can get a slip of his/her transaction made.

## ➤ LIST OF CLASSES, INTERFACES & ENUMS:

### 1. **ATMMain:**

- The main class of ATM project.
- Has Instances of other classes.
- Main method takes input from the user/admin and all core operations are abstracted here.

### 2. **Display Screen: (*extends DataBase class*)**

- Stores methods which are responsible to display and convey information from ATM to screen.

### 3. **DataBase Class:**

- Stores ATM related Info, Methods and acts as a medium for User Info by instantiating UserInfo class and fetches user info from txt file.
- **Encapsulates** ATM's Data.

### 4. **UserInfo Class: (*implements ValidChanges*)**

- **Encapsulates** Users' Data.
- Stores Users' related Data privately.
- Has Getter and Setter Methods declared in ValidChanges Interface.

### 5. **Transaction Class: (*implements Withdraw and Deposit interfaces*)**

- Has implemented methods to perform withdraw and deposit related transactions i.e. calculates and updates amount to be withdrawn or deposited.

### 6. **Admin Class:**

- Has Admin's Username and Password **ENUMS** stored privately.
- Has check methods to check UserName and Password.

### 7. **Withdraw Interface:**

- Max amount to be withdrawn per transaction is set here.
- Has method which is responsible to calculate the denomination wise amount to be given to the

customer as requested, which is defined by Transaction class.

#### 8. Deposit Interface:

→ Has method which is responsible to collect the denomination wise amount from customer, defined by transaction class.

#### 9. Valid Changes Interface:

→ Has declared setter and getter methods to change and update only some of the users' data but not all, as set by ABC Bank.

10. **AccType Enum:** Has different account types ex: savings, current.

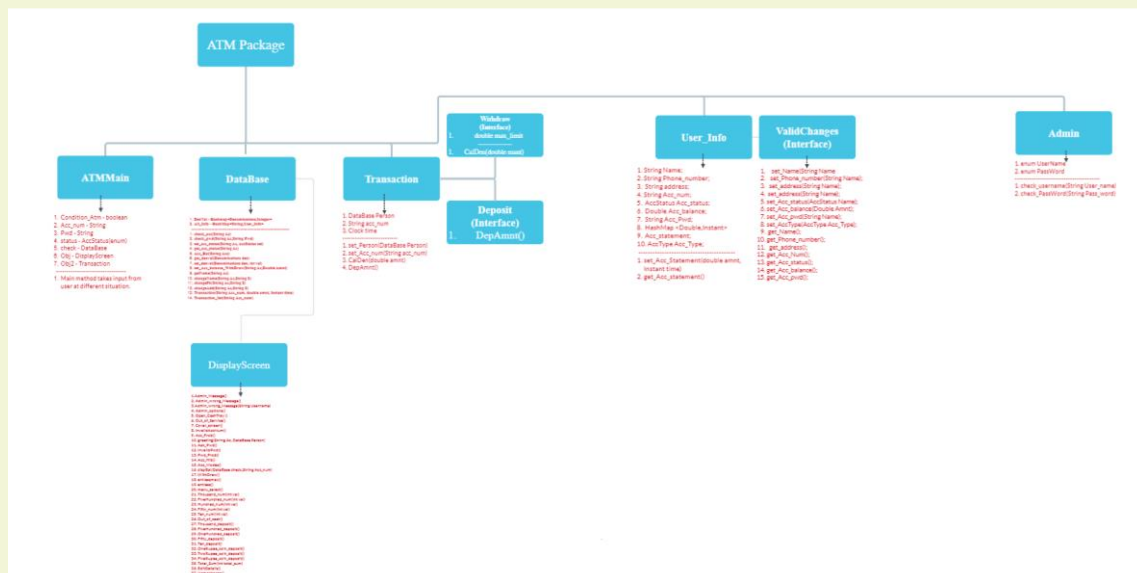
11. **AccStatus Enum:** Has different account modes like IN\_USE, BLOCKED, HIBERNATE etc.

12. **Denominations Enum:** Has different Denominations Values like Fifty, One\_hundered, Five\_hundred, One\_thousand etc.

13. **Transaction Enum:** Has different transaction types which are Withdraw and deposit.

14. **Input txt File:** Has User's Information.

## TREE DIAGRAM:



### **Encapsulation**: (An Example in the code)

UserInfo class hides the data of customers by using the access modifier PRIVATE.

---

### **Abstraction**: (An Example in the code)

Interface Valid Changes contains methods which are abstract and abstraction is used here because what to modify in user data is bank specific, therefore whatever the personal details the bank assumed the customer can modify is enclosed abstractly in Valid Changes Interface.

---

### **Inheritance**: (An Example in the code)

Class DisplayScreen extends the DataBase class to simplify getting Atm and Users' data from that class.

---

### **Polymorphism**: (An Example in the code)

Admin\_wrong\_message() and Admin\_wrong\_message(String UserName) in Display class are overloaded.