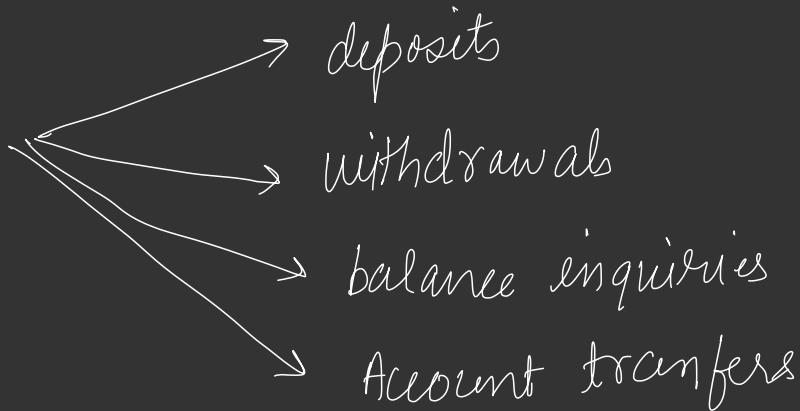




## The ATM System

→ An ATM allows a bank customer to perform financial transactions remote without the need for a teller or bank branch.

Transactions include



To ATM → we need a bank card., use account must use  
PIN (personal identification number)

## Expectations from Interviewee

- What are components of an ATM?
- Is the ATM necessarily placed inside a room?
- Does an ATM have a fingerprint scanner?
- What is the withdrawal limit of an ATM?
- Can we check our account balance using an ATM?
- Can we set a PIN using an ATM?
- Can we set a limit for withdrawal by the user?
- What happens when the amount entered by the user for withdrawal is greater than the user's account balance?
- What happens when the amount entered by the user for withdrawal is greater than the ATM cash limit?
- What happens when the amount entered by the user exceeds the total cash present in the ATM?
- Can ATM be used for online transaction?

## Requirement Collection

R1: Each user has a single account at the bank that they can access by inserting their card into the ATM.

R2: The main components of the ATM system that facilitate interactions between the user & the machine are listed.

- Card Reader → To read the user's ATM card
- Keypad → To enter information such as the user's PIN
- Screen → To display message to the user, such as prompt or error message
- Cash Dispenser → To dispense cash to the user
- Printer → To print receipt for the user.
- Network Infrastructure → To connect with the bank's computer system in order to access account information & complete transaction.

R3: The ATM system must authenticate the user based on the PIN they enter to ensure that only authorized users can access their accounts.

R4: — All transactions are possible after the successful authentication of the ATM Card

R5: — The user can have 2 types of accounts - Current & Savings - and can perform the following operation

- Balance inquiry
- Cash withdrawal
- Funds / money transfer

R6: — At the end of a transaction, the user has the option to start another transaction or end their session.

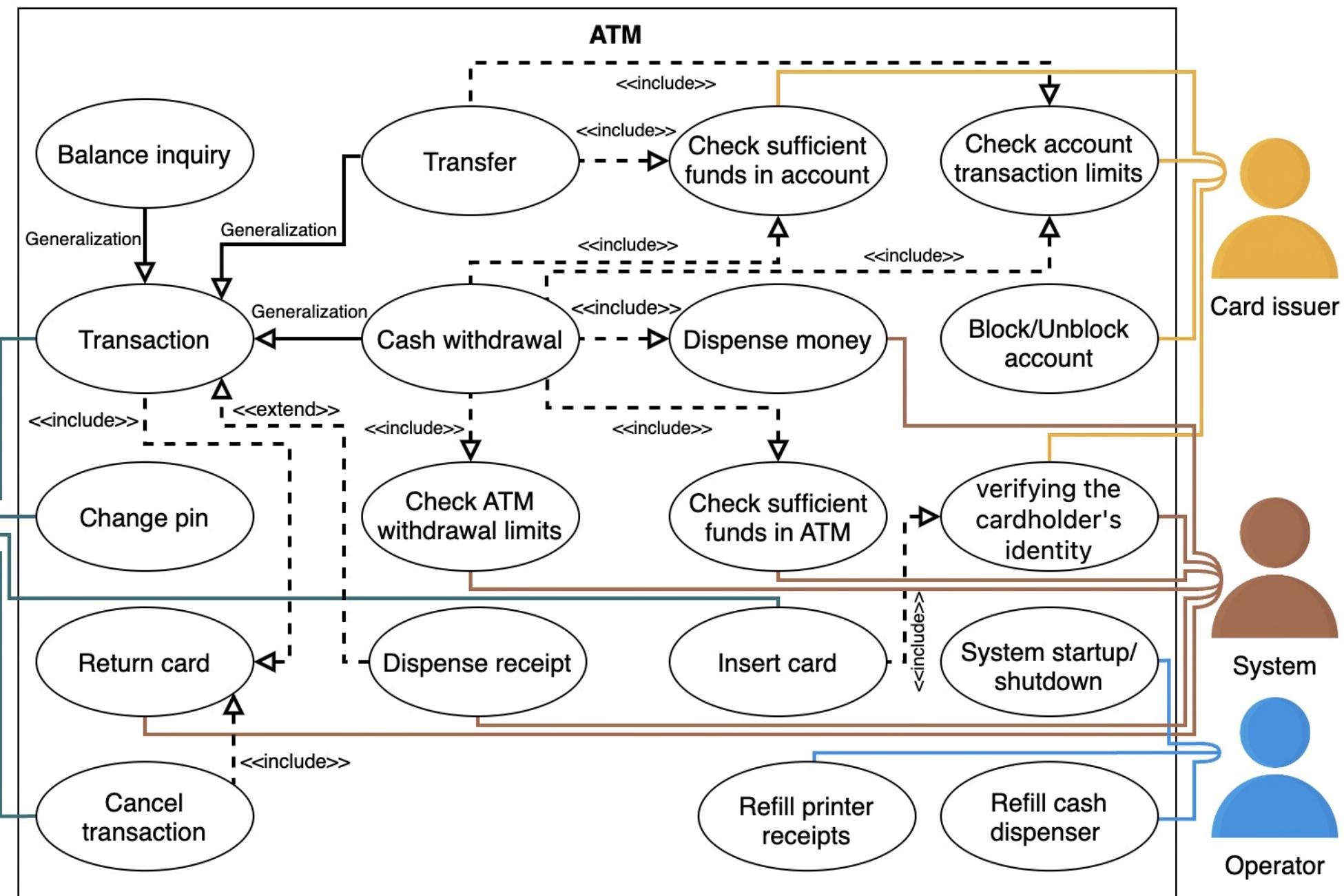
## Actors

Primary Actors : Card holders

Secondary Actors : Card issuer  
System  
Operator

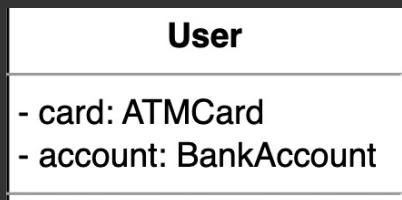
Cardholder	Card issuer	System	Operator
Change PIN	Check for sufficient funds in account	verifying the cardholder's identity	System startup/shutdown
Insert card	Check cardholder account transection limits	Check ATM withdraw limits	Refill printer receipts
Cancel transaction	Block/Unblock account	Check for sufficient funds in ATM	Refill cash dispenser
Transaction	Verify the cardholder's identity	Return card	
		Dispense money	
		Dispense receipt	

# Use Case Diagram

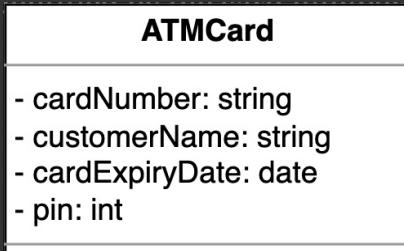


# Class Diagram

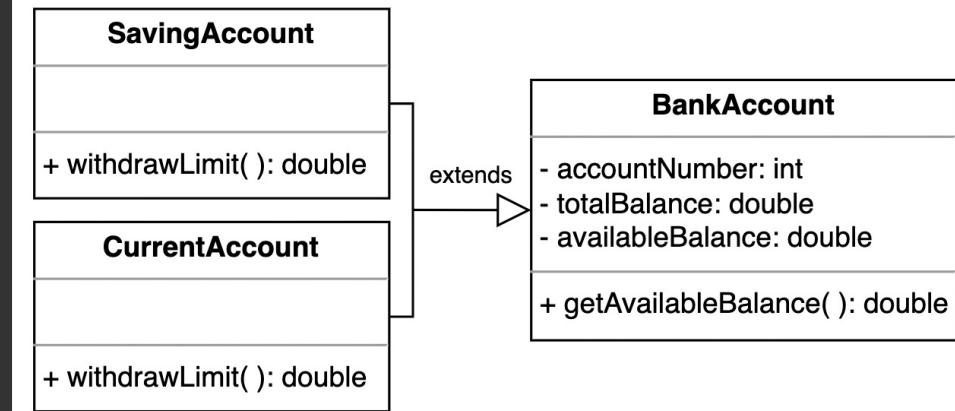
1. User



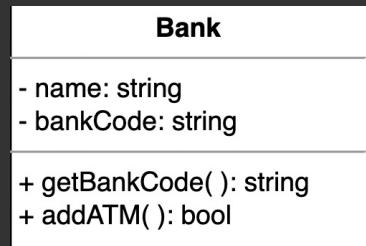
2. ATM Card



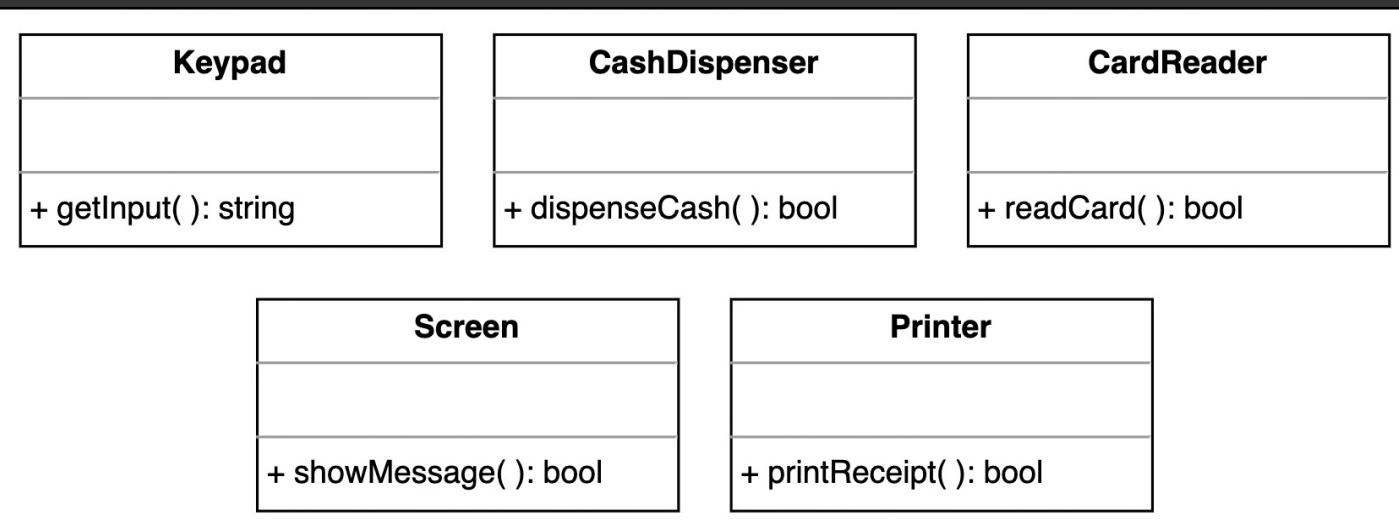
3. Bank Account



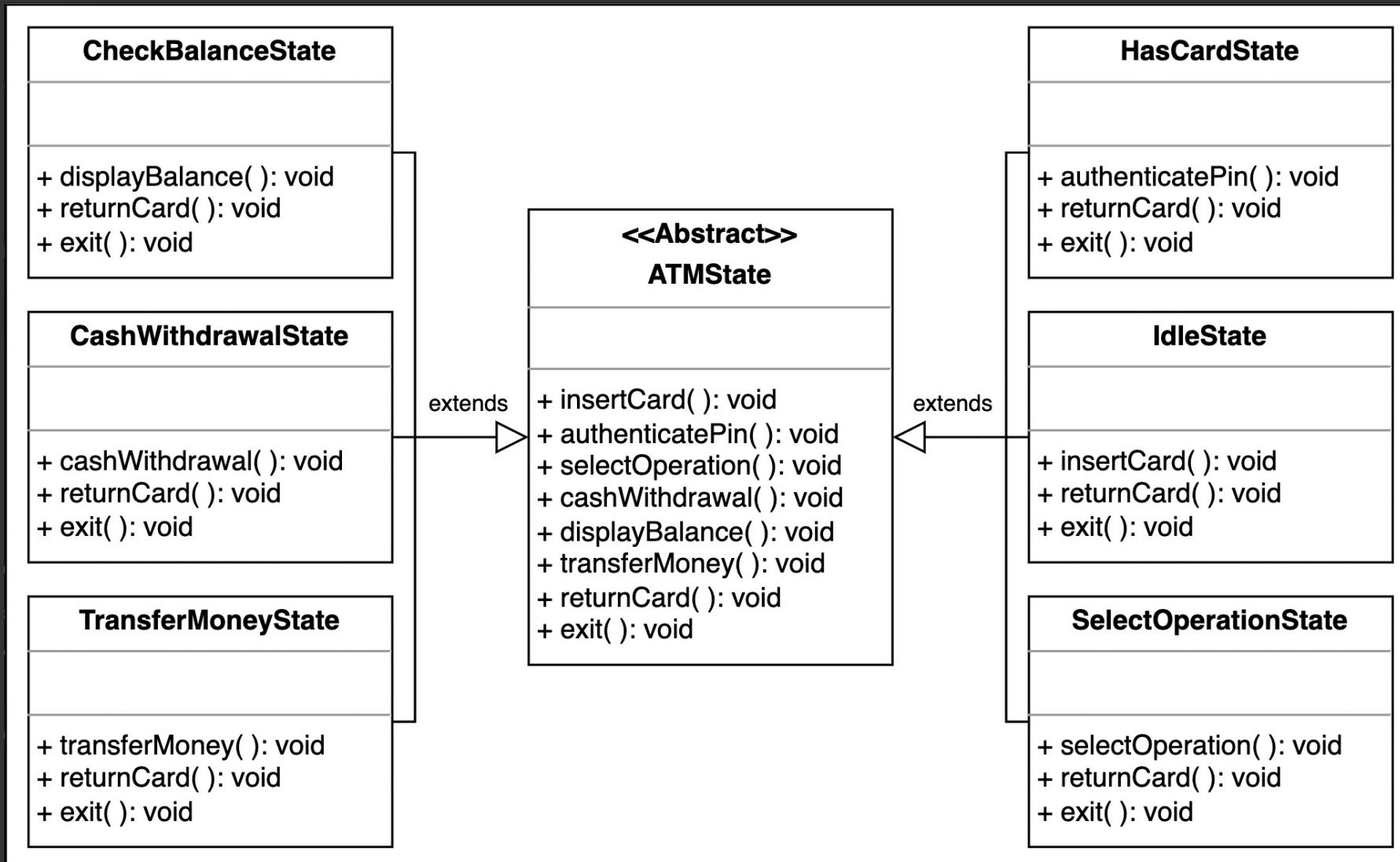
4. Bank



5. Card Reader, Cash dispenser, Keypad, Screen & printer



## 6. ATM State



## 7. ATM

### **ATM**

- atmObj: ATM  
- currentState: ATMState  
- atmBalance: float  
- noOfHundredDollarBills: int  
- noOfTwentyDollarBills: int  
- noOfTwoDollarBills: int

+ displayCurrentState( ): void  
+ initializeATM( ): void

## 8. ATM Room

### **ATMRoom**

- atm: ATM  
- user: User

## 9. Enumerations

### **<<enumeration>>**

#### **ATMState**

Idle  
HasCard  
SelectionOption  
Withdraw  
TransferMoney  
BalanceInquiry

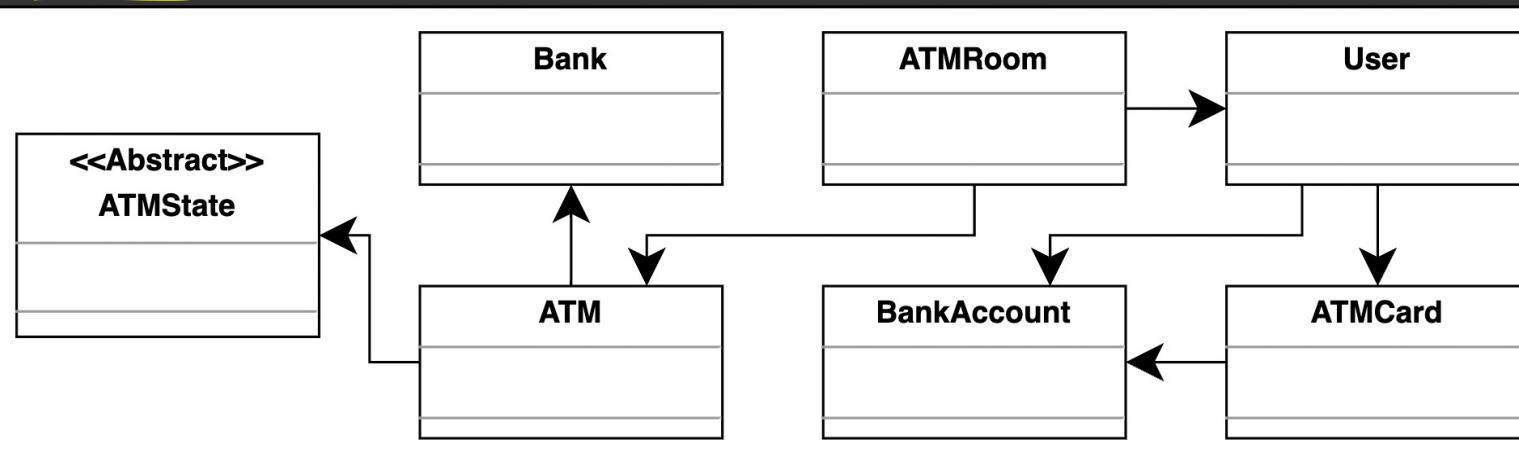
### **<<enumeration>>**

#### **TransactionType**

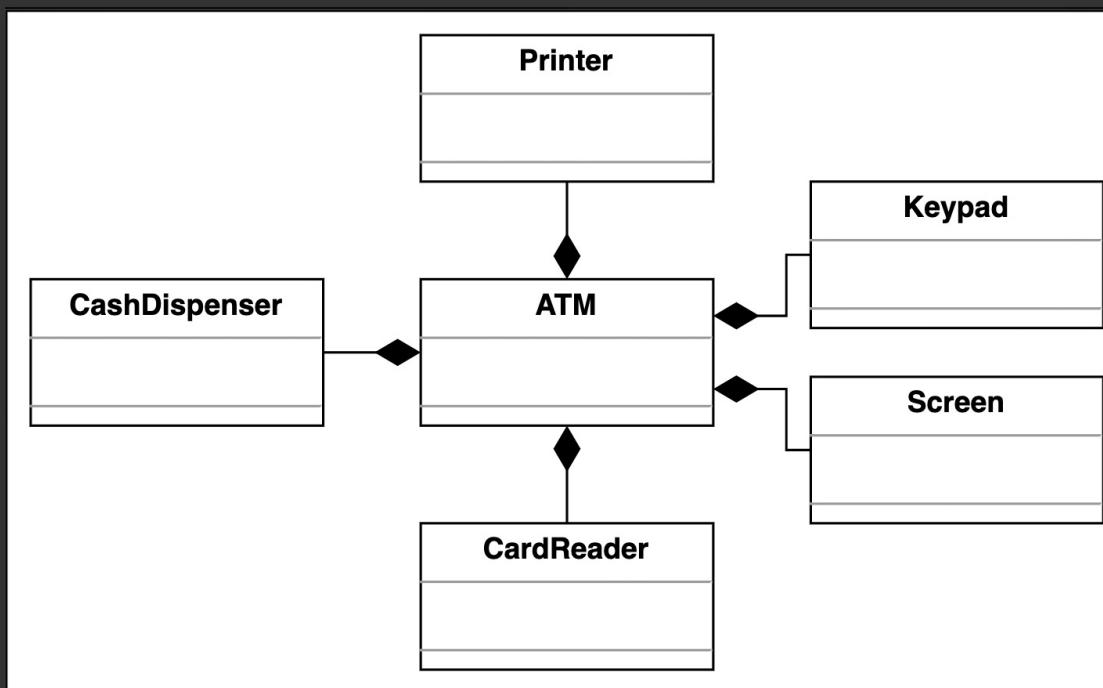
BalanceInquiry  
Withdraw  
Transfer

## Relationship between classes

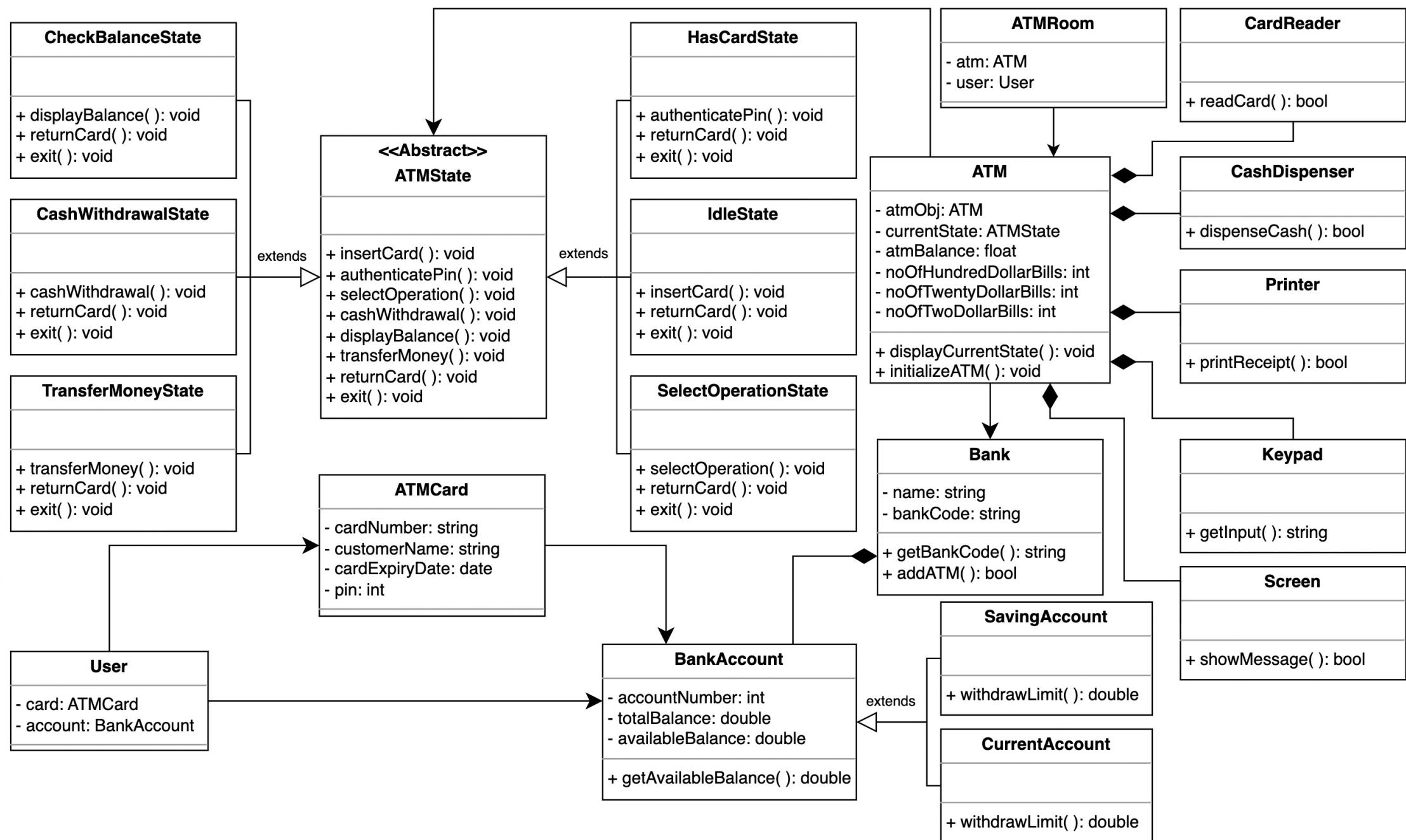
### Association



### Composition

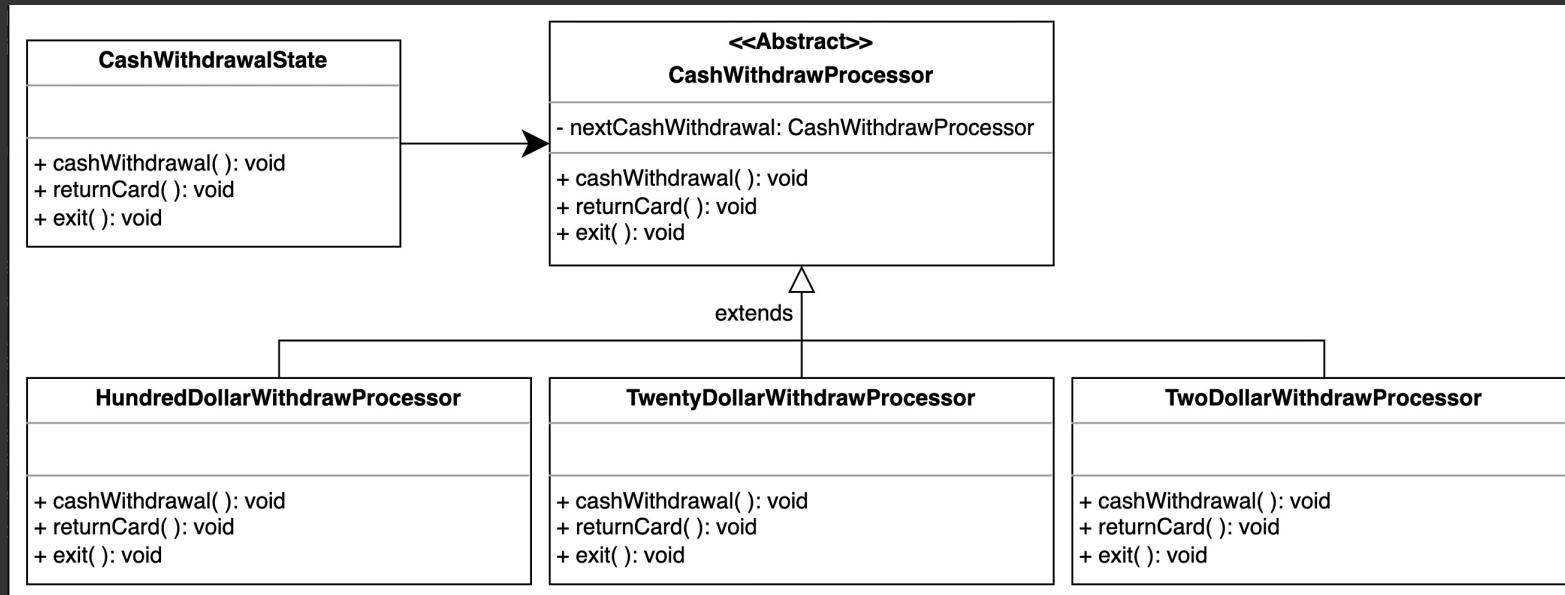


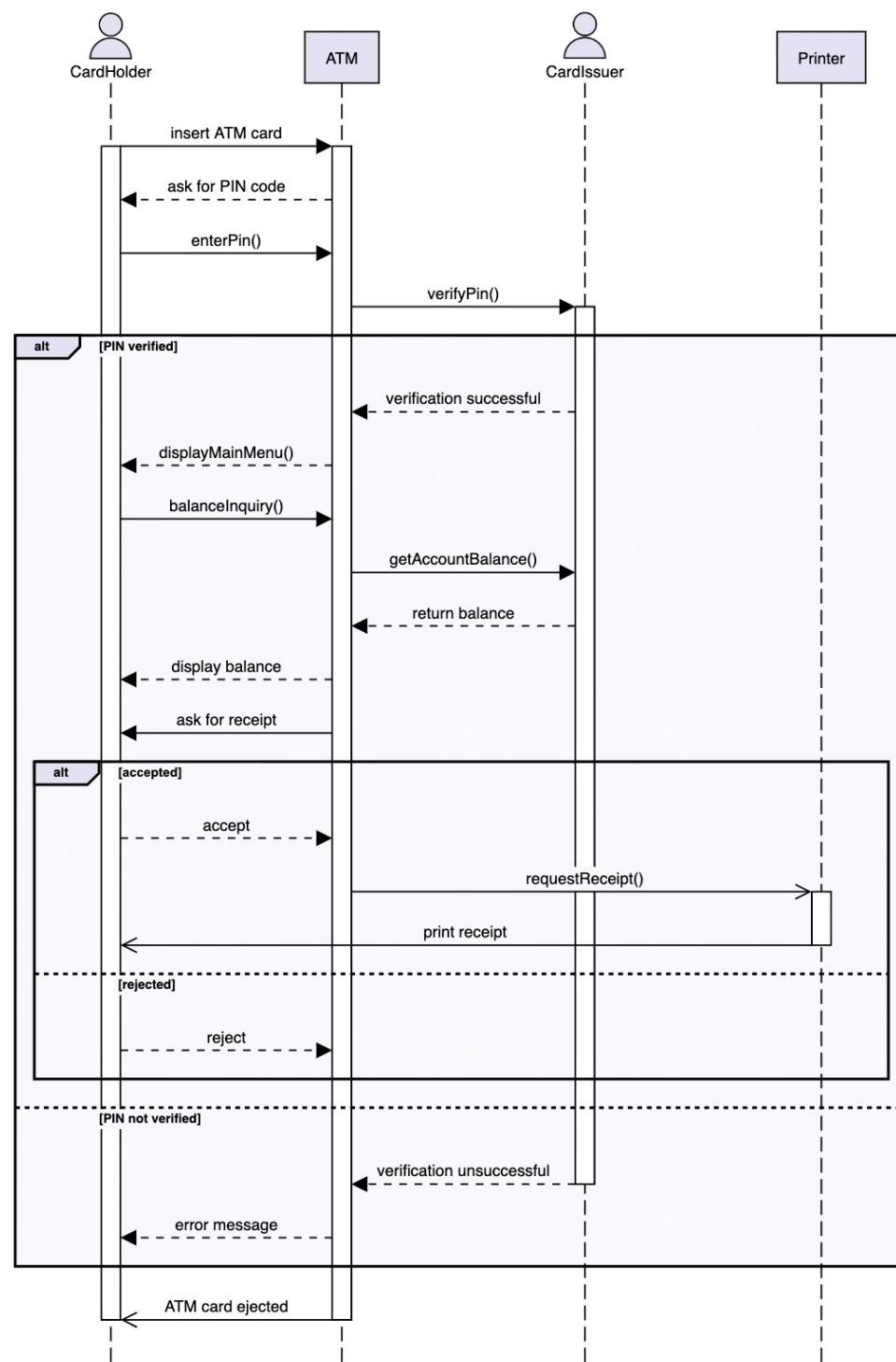
# Class Diagram for ATM System



## Additional Requirement

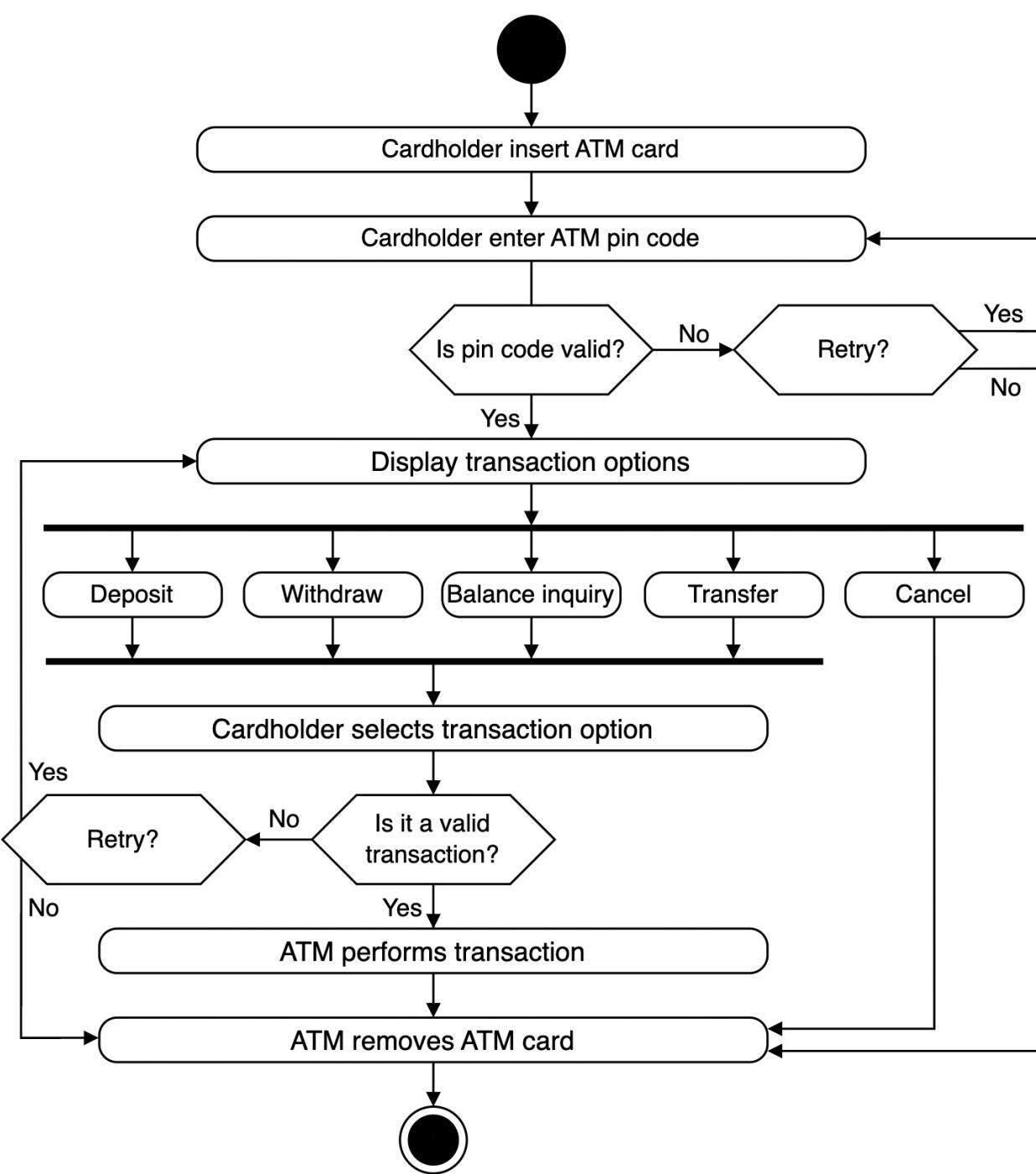
We need to withdraw in combination of → hundred, twenty & 2 dollar

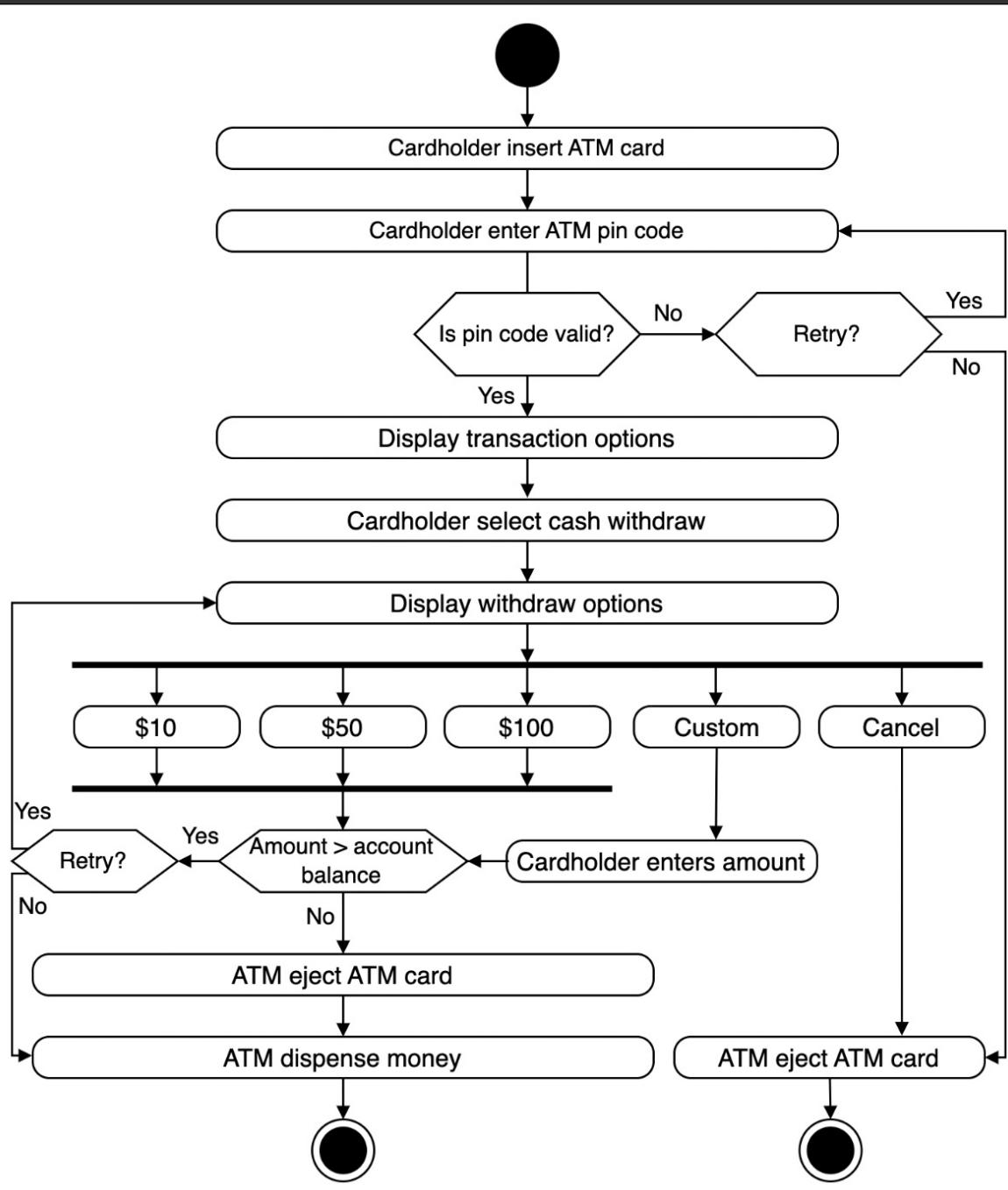




→ Sequence diagram of balance inquiry

# Activity Diagram of ATM System





## Code of ATM System

1. Enumeration

```
// Enumeration
enum ATMState {
    Idle,
    HasCard,
    SelectionOption,
    Withdraw,
    TransferMoney,
    BalanceInquiry
}
```

2. User & ATMCard

```
public class User {
    private ATMCard card;
    private BankAccount account;
}

public class ATMCard {
    private String cardNumber;
    private String customerName;
    private Date cardExpiryDate;
    private int pin;
}
```

3. Bank & Bank Account

```
public class Bank {
    private String name;
    private String bankCode;

    public String getBankCode();
    public boolean addATM();
}

public class BankAccount {
    private int accountNumber;
    private double totalBalance;
    private double availableBalance;

    public double getAvailableBalance();
}

public class SavingAccount extends
BankAccount {
    public double withdrawLimit();
}

public class CurrentAccount extends
BankAccount {
    public double withdrawLimit();
}
```

4. Card Reader, Card dispenser, printer, screen & Keypad

```
public class CardReader {
    public boolean readCard();
}

public class CashDispenser {
    public boolean dispenseCash();
}

public class Keypad {
    public String getInput();
}

public class Screen {
    public void showMessage();
}

public class Printer {
    public void printReceipt();
}
```

## 5. ATM State

```
public abstract class ATMState {  
    public abstract void insertCard(ATM atm, ATMCARD card);  
    public abstract void authenticatePin(ATM atm, ATMCARD card, int pin);  
    public abstract void selectOperation(ATM atm, ATMCARD card, TransactionType tType);  
    public abstract void cashWithdrawal(ATM atm, ATMCARD card, int withdrawAmount);  
    public abstract void displayBalance(ATM atm, ATMCARD card);  
    public abstract void transferMoney(ATM atm, ATMCARD card, int accountNumber, int transferAmount);  
    public abstract void returnCard();  
    public abstract void exit(ATM atm);  
}  
  
public class IdleState extends ATMState {  
    @Override  
    public void insertCard(ATM atm, ATMCARD card) {  
        // definition  
    }  
  
    @Override  
    public void returnCard() {  
        // definition  
    }  
  
    @Override  
    public void exit(ATM atm) {  
        // definition  
    }  
}
```

```
public class HasCardState extends ATMState {  
    @Override  
    public void authenticatePin(ATM atm, ATMCARD card, int pin) {  
        // definition  
    }  
  
    @Override  
    public void returnCard() {  
        // definition  
    }  
  
    @Override  
    public void exit(ATM atm) {  
        // definition  
    }  
}  
  
public class SelectOperationState extends ATMState {  
    @Override  
    public void selectOperation(ATM atm, ATMCARD card, TransactionType tType) {  
        // definition  
    }  
  
    @Override  
    public void returnCard() {  
        // definition  
    }  
  
    @Override  
    public void exit(ATM atm) {  
        // definition  
    }  
}
```

```
public class CheckBalanceState extends ATMState {  
    @Override  
    public void displayBalance(ATM atm, ATMCard card) {  
        // definition  
    }  
  
    @Override  
    public void returnCard() {  
        // definition  
    }  
  
    @Override  
    public void exit(ATM atm) {  
        // definition  
    }  
}  
  
public class CashWithdrawalState extends ATMState {  
    @Override  
    public void cashWithdrawal(ATM atm, ATMCard card, int withdrawAmount) {  
        // definition  
    }  
  
    @Override  
    public void returnCard() {  
        // definition  
    }  
  
    @Override  
    public void exit(ATM atm) {  
        // definition  
    }  
}
```

```
public class TransferMoneyState  
extends ATMState {  
    @Override  
    public void transferMoney(ATM atm,  
    ATMCard card, int accountNumber, int  
    transferAmount) {  
        // definition  
    }  
  
    @Override  
    public void returnCard() {  
        // definition  
    }  
  
    @Override  
    public void exit(ATM atm) {  
        // definition  
    }  
}
```

## 6. ATM & ATM Room

```
public class ATM {  
    private static ATM atmObject = new ATM(); //Singleton  
    private ATMState currentATMState;  
    private int atmBalance;  
    private int noOfHundredDollarBills;  
    private int noOfFiftyDollarBills;  
    private int noOfTenDollarBills;  
  
    // References to various ATM components  
    private CardReader cardReader;  
    private CashDispenser cashDispenser;  
    private Keypad keypad;  
    private Screen screen;  
    private Printer printer;  
  
    public void displayCurrentState();  
    public void initializeATM(int atmBalance, int noOfHundredDollarBills, int  
noOfFiftyDollarBills, int noOfTenDollarBills);  
}  
  
public class ATMRoom {  
    private ATM atm;  
    private User user;  
}
```