**ATM Class Diagram**

A brief description of the model:

The bank creates a customer by creating a bank Account with the customers data and a starting amount of money or None  
then it will issue a card with the Account also it’s possible to add another card to an existing account and add another account to an existing customer

The bank also creates ATMs and give them numbers  
a customer through an ATM can make a transaction with the card by inputting the amount, type of transaction ,cardNo and PinCode which is verified by the card reader   
the card reader also reads the Account number from the magnetic strip of the card  
the ATM initialize the transaction which is processed to the bank account and stored in the Transaction record  
the transaction record also holds the number of the ATM

**Classes:**

* **Customer Class**Attributes

- name: string   
- customerID:int   
- bankAccounts: ArrayList<BankAccount>   
- contactinfo:int   
-cards:ArrayList<Card>   
-PinNo:int  
   
Methods:

+ Customer()

+getName():String

+getCustomerID:int

+addCard():void

+addBankAcc():void

+getCards():ArrayList<Card>

+getAllBankAccs():ArrayList<BankAcc>

+getSpecificCard:Card

+ makeTransaction()

* **Card Class**

Attributes  
+ Customer()

+getName():String

+getCustomerID:int

+addCard():void

+addBankAcc():void

+getCards():ArrayList<Card>

+getAllBankAccs():ArrayList<BankAcc>

+getSpecificCard:Card

+ makeTransaction()  
  
Methods  
+ Card()

+ getCardNo():int

+ getBankAccNo:int

+ getPinCode:int

* **ATM class**Attributes  
  + ATM\_No:int

+ bankName:string   
Methods  
+ processTransaction()

- cardReader(): BankAccount

-verifyCard():boolean

* **Transaction Class**Attributes  
  - transactionInfo:int

-transactionAmount:float

-transactionType:String

-bankAcc:BankAccount

-atmNo:int  
  
Methods  
+getTransactionInfo():String

+Transaction()

+withdraw():void

+deposit():void

* **Bank account Class**Attributes  
  - BankAccNO:int

+ customer:Customer

- customerID:int

- creditAmount:int

- transactionRecords: ArrayList<Transaction>

- cardsConnected: ArrayList<Cards>  
Methods  
+ BankAccount()

+ addCard():void

+ getCreditAmout():float

+ getBankAccNo():int

+ getTransactionRecord(): ArrayList<Transaction>

+ setCreditAmount():void

* **Bank Class**Attributes  
  + BankName:string

-allATMs:ArrayList<ATM>

-allCustomers: <ArrayList<Customer>

-bankAccounts: ArrayList<BankAccount>  
Methods  
+Bank()

+ AddCards()

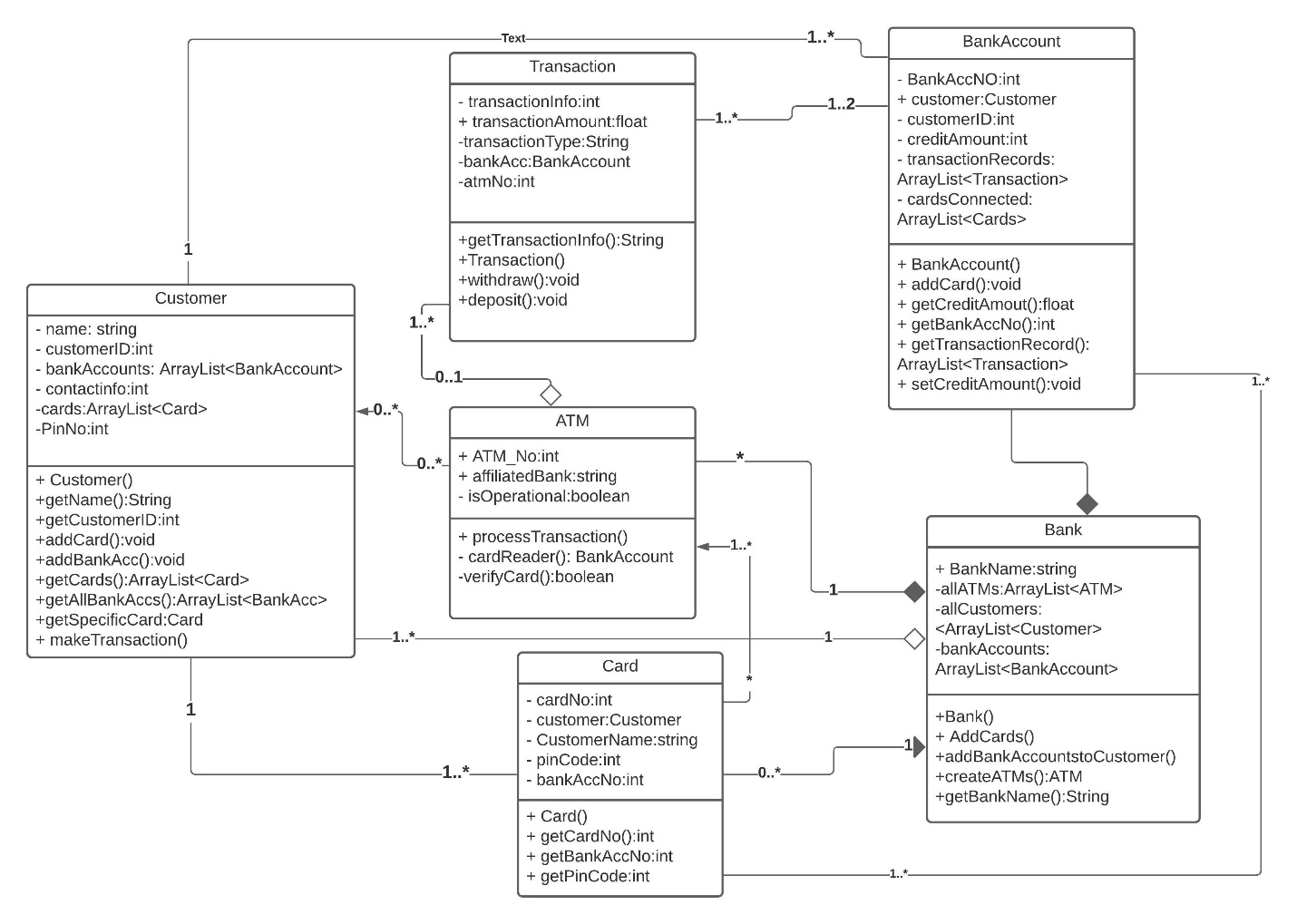
+addBankAccountstoCustomer()

+createATMs():ATM

+getBankName():String

**Associations**

* **Customer – Card**bidirectional   
  customer holds and uses cards  
  card has holder name on the back of the card and bank account number on the magnetic strip  
  cardinality ratio: 1---1..\*  
  a card must have an holder   
  a customer can hold one or multiple cards
* **Customer – Bank Account**  
  bidirectional  
  customer have a bank account  
  bank account is owned by a customer  
  cardinality ratio: 1---1..\*  
  a customer can have 1 or many bank accounts  
  a bank account is owned by 1 customer
* **Customer – ATM**unidirectional  
  customer makes transaction via ATM  
  cardinality ratio: 0..\*---0..\*  
  a customer can use zero or many ATMs  
  an ATM can be used by zero many customers
* **Card – Bank Account**bidirectional  
  card has bank account number on its magnetic strip  
  bank account has list of cards connected to it  
  cardinality ratio: 1..\*---1  
  a card have 1 account connected to it  
  a bank account can have multiple cards connected to it
* **ATM – Transactions**aggregation  
  transactions are a component of ATM but they still can be made without ATMs  
   ATMs processes transaction with its types  
  cardinality ratio: 0..1---1..\*  
  a transaction is processed in zero or 1 ATM  
  an ATM will make at least 1 transaction
* **ATM – Bank**composition  
  an ATM can’t exist without a bank  
  bidirectional  
  ATM is affiliated to a bank  
  the bank distributes affiliated ATMs  
  cardinality ratio: \*---1  
  an ATM must be affiliated to 1 bank  
  a bank can have many ATMs
* **Transaction – Bank Account**Bidirectional  
  transaction info contains the involved bank accounts  
  bank accounts has a record of all transactions involving it  
  cardinality ratio: 1..\*---1  
  a transaction involves 1 bank accounts  
  a bank account will at least have 1 transaction(opening day)

****