ATM DESIGN REPORT

BY RIFA SAFEER SHAH AND NATHANIEL MONTE DE RAMOS

Table of Contents

Request for Proposal

Date of Issuance: February 26, 2020

PROJECT INFORMATION

This Request for Proposal is issued to design and build a software system for an ATM machine. Each customer can have only one account at the bank.

PROGRAM IDEA

The system should display an interface to the user like a welcome screen, “Please input some info (card and pin)”. There are cases where the pin is wrong, there is no cash in the ATM machine, and other such cases. The system should provide the user their requested funds if available in their account and accept funds as deposits. Error messages for unreadable and invalid cards.

Can be used as a desktop program.

REQUIREMENTS

1. Display screen with instructions for customer
2. Card Reader
3. Keypad
4. Pin Verify
5. Cash Dispenser
6. Withdraw
    - The user has the ability to take away money from his/her bank account
7. Deposit
    - The user has the ability to put money to his/her bank account
8. Check Balance
    - The user has the ability to check the amount of money on his/her bank account
9. Transfer Fund
    - The user can transfer money from one account to another
10. Exit
    - The user will exit the program

PROJECT TEAM

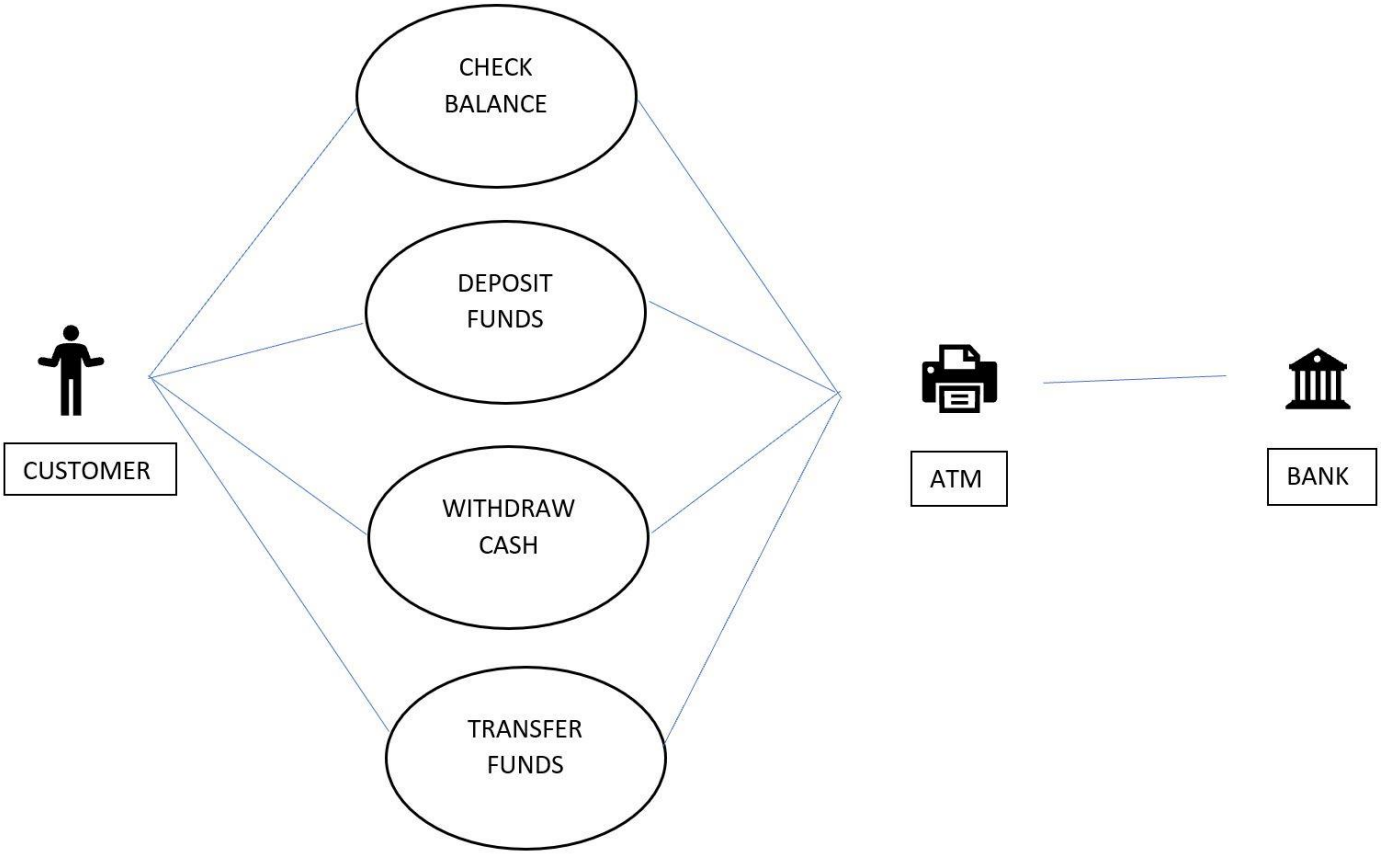
Nathaniel Monte De Ramos

Rifa Safeer Shah

Use Cases and Scenarios

Date of Issuance: February 26, 2020

USE CASE DIAGRAM



CASE I

Use case name: Deposit

Summary: Customer deposits cash or check to their bank account using an ATM machine with a debit card.

Actor: Customer of the Bank

Precondition:

The user has a valid debit card and an active account.

Description (Basic flow of events):

1. The customer inserts their debit card into the card reader on the ATM machine.
2. The ATM machine reads the account information and checks if the card and pin is valid.
3. If the debit card and the bank account is valid then the machine prompts the user to select one of the four transaction options (Withdraw, Transfer Funds, Check Balance, Deposit).
4. The customer selects Deposit transaction.

5. The ATM machine asks the user how much money to deposit.
6. The customer enters the amount to be deposited.
7. The customer inserts the check or cash into the deposit input.
8. The ATM machine prints the receipt for the transaction.
9. The use case ends, the card is returned to user and the system returns to the home screen.

Alternative flows:

1. A. The card is reported stolen.
1. B. Incorrect PIN number is entered.
2. Account is frozen.
8. Debit card is stuck in the card reader.

Postcondition:

The money will be deposited to the account successfully and the program will return to the home screen.

Non-functional requirements:

Machine shall deposit the correct amount of money in at least 99% of cash or check deposit.

CASE II

Use case name: Withdraw

Summary: Customer withdraws cash from the ATM machine with a debit card and an active bank account.

Actor: Customer of the Bank

Precondition:

The user has a valid debit card and an active account.

Description (Basic flow of events):

1. The customer inserts their debit card into the reader on the ATM machine.
2. The ATM machine reads the account information of the customer.
3. The system identifies the customer and authenticates if the debit is valid by asking the customer for the pin number.
4. Customer enters the pin number.
5. The machine links the debit card to the bank account.
6. If the debit card and the bank account is valid then the machine prompts the user to select one of the four transaction options (Withdraw, Transfer Funds, Check Balance, Deposit).
7. The customer selects Withdraw transaction.
8. The machine prompts user to enter the amount to withdraw from the ATM machine.
9. The machine checks if the machine and the bank account have enough funds.
10. The machine releases the requested cash.
11. The machine ejects the debit card after cash is taken.
12. The customer has successfully withdrawn the required cash.
13. The user is asked if they want a receipt.
14. The Use Case ends, and the system displays home screen.

Alternative flows:

1. Debit card is not readable.
2. A. Account is frozen.

4. Incorrect PIN number is entered.
10. A. Not enough money in the ATM machine.
10. B. Not enough money in the Bank account.
11. Customer forgot to take the money from the tray.
12. Debit card is stuck in the card reader.

Postcondition:

Customer gets his/her money and machine will go back to its home interface.

Non-functional Requirements:

Machine shall give the correct amount of cash in at least 99% of cash withdrawal.

CASE III

Use case name: Check Balance

Summary: Customer can check on how much money is left in his/her account.

Actor: Customer of the Bank

Precondition:

The user has a valid debit card and an active account.

Description (Basic flow of events):

1. The customer inserts their debit card into the card reader on the ATM machine.
2. The ATM machine reads the account information of the customer.
3. The system identifies the customer and authenticates if the debit is valid by asking the customer for the PIN number.
4. Customer enters the PIN number.
5. The machine links the debit card to the bank account.
6. If the debit card and bank account is valid then the machine prompts the customer to select one of the four services (Withdraw, Transfer Funds, Check Balance, Deposit).
7. The customer selects the Check Balance option.
8. The machine will show how much balance is in the account.
9. The machine has an exit option that will be pressed by the customer.
10. The machine ejects the debit card after the customer pressed the exit option.
11. The Use case ends, and the system displays the home screen.

Alternative flows:

1. A. Debit card is stuck in the card reader.
2. B. Debit card is not readable.
3. The card is reported stolen.
4. Account is frozen.
5. Customer does not respond halfway through the process.
6. Incorrect PIN entered.

Postcondition:

The system goes back to the home page.

Non-functional Requirement:

Machine will show account balance as long there is no problem with the account.

CASE IV

Use case Name: Transfer Fund

Summary: Customer transfers money from one account to another.

Actor: Customer of the Bank

Precondition:

The user has a valid debit card and an active account.

Description (Basic flow of events):

1. The customer inserts their debit card into the reader on the ATM machine.
2. The ATM machine reads the account information of the customer.
3. The system identifies the customer and authenticates if the debt is valid by asking the customer for the PIN number.
4. Customer enters the PIN number.
5. The machine links the debit card to the bank account.
6. If the debit card and the bank account are valid then the machine prompts the user to select from one of the four options. (Withdraw, Deposit, Check Balance, Transfer Fund).
7. The customer selects Transfer Fund option.
8. The machine prompts user to enter the amount needed to transfer.
9. The machine checks if the machine and the bank account have enough funds.
10. The machine deducts the account of the requested cash.
11. The machine ejects the debit card after cash is taken.

Alternative flows:

1. Debit card is not readable.
2. A. Account is frozen.
4. Incorrect PIN number is entered.
10. A. Not enough money in the ATM machine.
10. B. Not enough money in the Bank Account.

Postcondition:

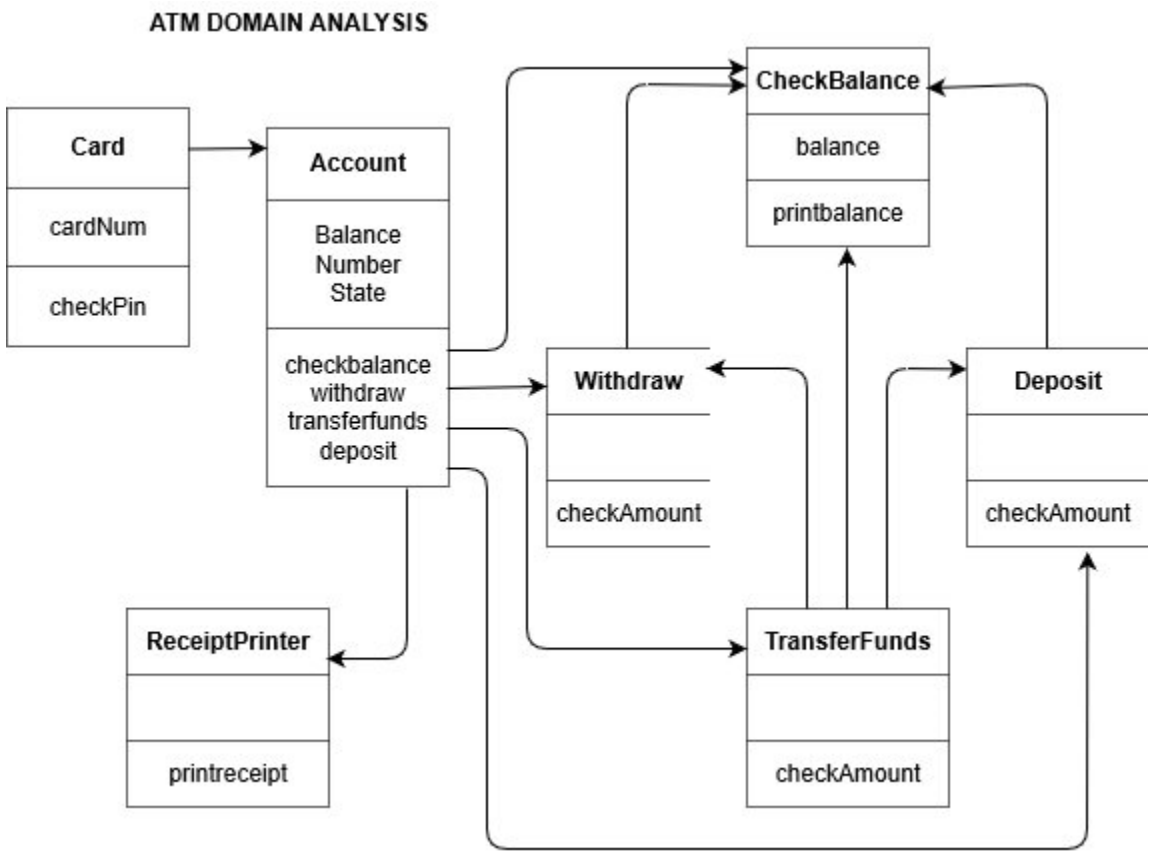
Customer get his/her money and machine will go back to its home interface.

Non-functional Requirements:

Machine shall transfer the correct amount of cash in at least 99% of Transfer process.

Domain Analysis

Date of Issuance: May 4, 2020



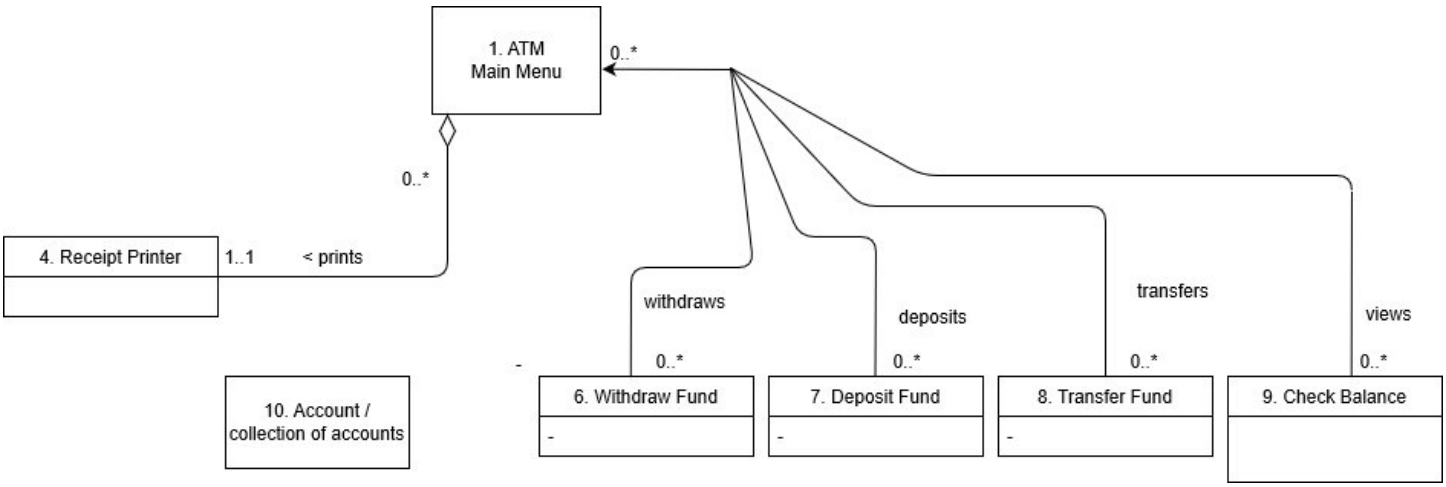
Class Diagram

Date of Issuance: March 9, 2020

Class Diagram Narrative

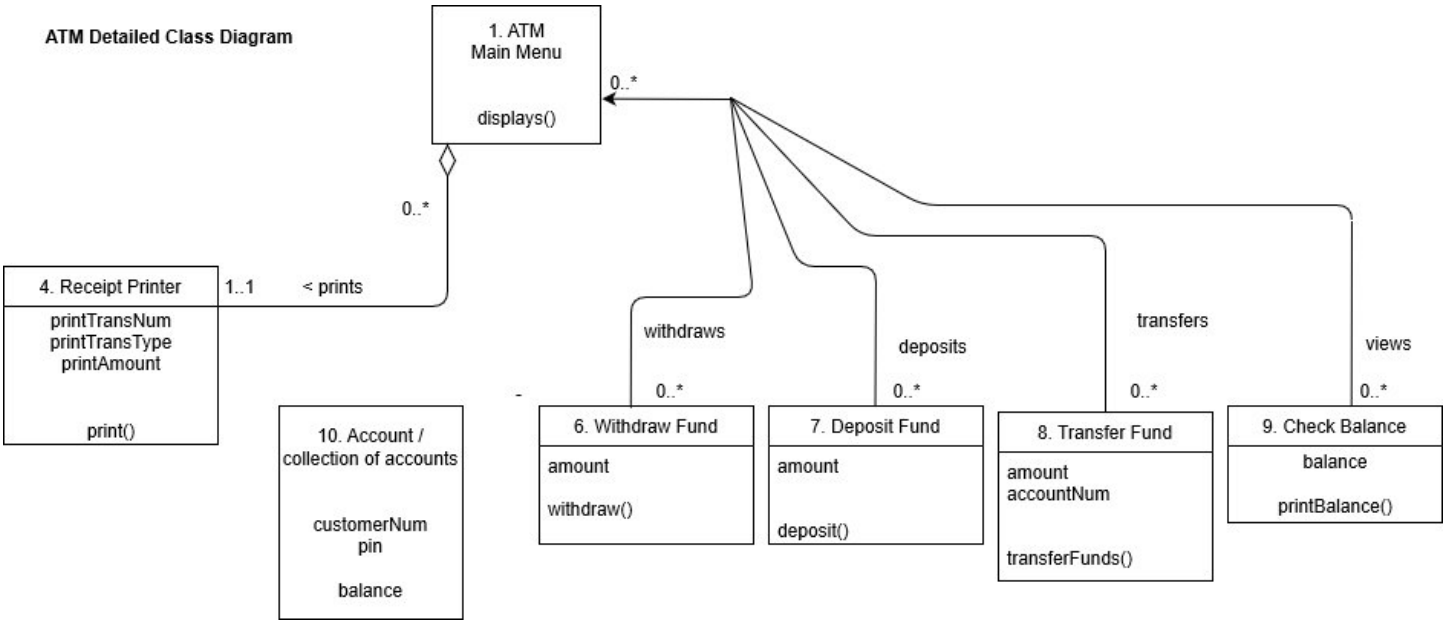
To make the class diagram we followed the use cases. The deposit class allows customers to deposit money into their accounts. To create the classes, we looked at the requirements of the project. The ATM displays four transactions that are Cash Deposit, Cash Withdrawal, Transfer Funds, and Check Balance. The CRC diagrams also helped us with the class diagram to make sure we included all the necessary aspects for the program. The customer accesses the ATM machine main menu through the display, fund dispenser, and receipt printer. The transactions are chosen from the main menu which allows the user to complete a task.

Class Diagram



Detailed Class Diagram

Date of Issuance: May 7, 2020



CRC

Date of Issuance: April 7, 2020

ATM	
This class is the main menu for the rest of the software. It works together with the collaborators to ensure the customer has the basic requirements when withdrawing or depositing money.	display fund dispenser receipt printer transaction

Check Balance	
This class knows the amount o money in the account. It also knows the account number and other information. It prints the amount of money in the account.	ATM

Deposit Funds	
This class allows customers to add money to their bank account. After the transaction a receipt will be printed.	check balance ATM

Account	
This class is a collection of bank accounts. It is where we will store the PIN number , account number, and money. It reads from a file and also writes to a file.	ATM

Receipt Printer	
This class handles printing the correct amount of funds in account at the end of the transaction.	ATM main menu

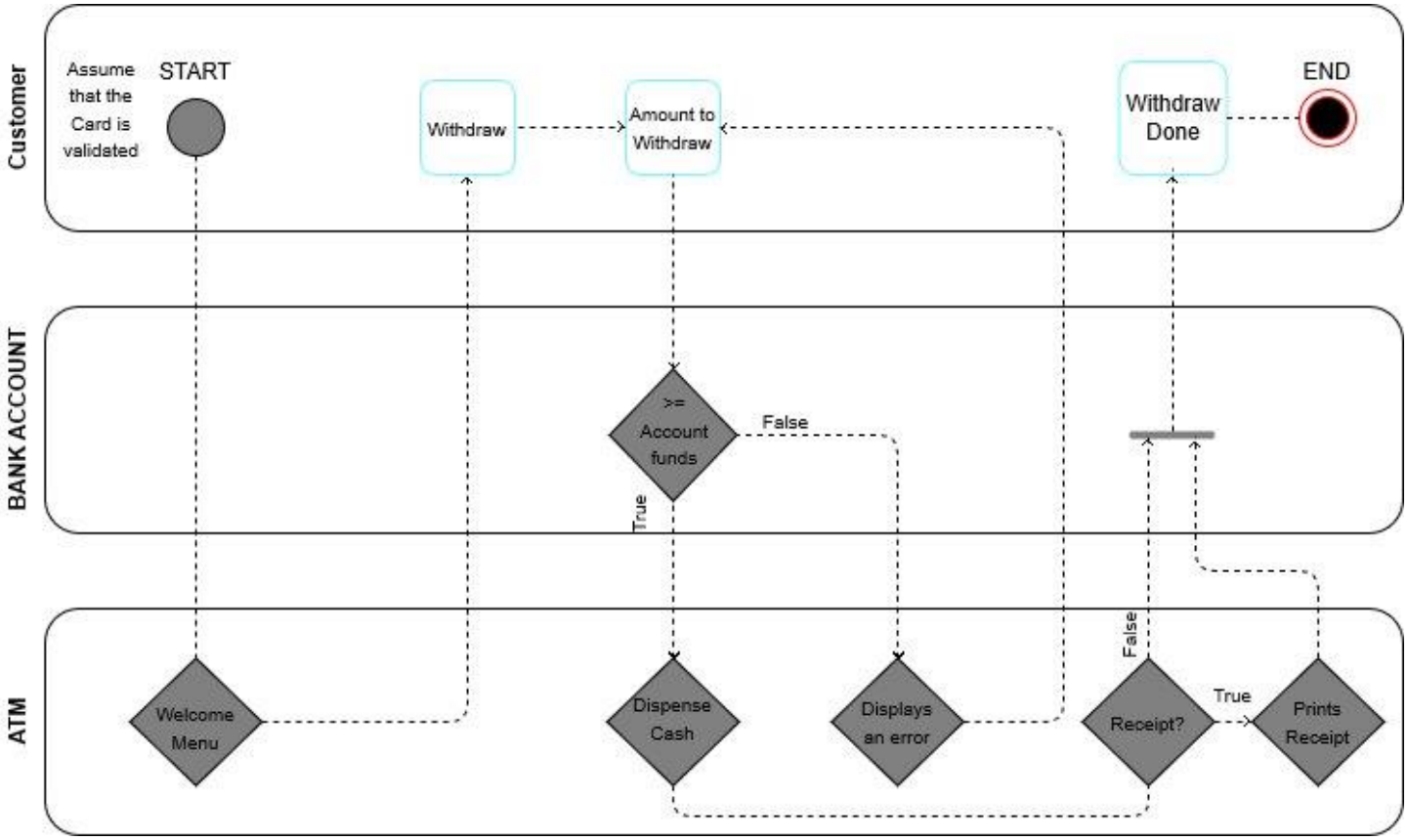
Transfer Funds	
This class knows the amount of money in the account. It has information about both accounts that are performing this transaction. It takes money from one account and adds it to the other account.	transaction

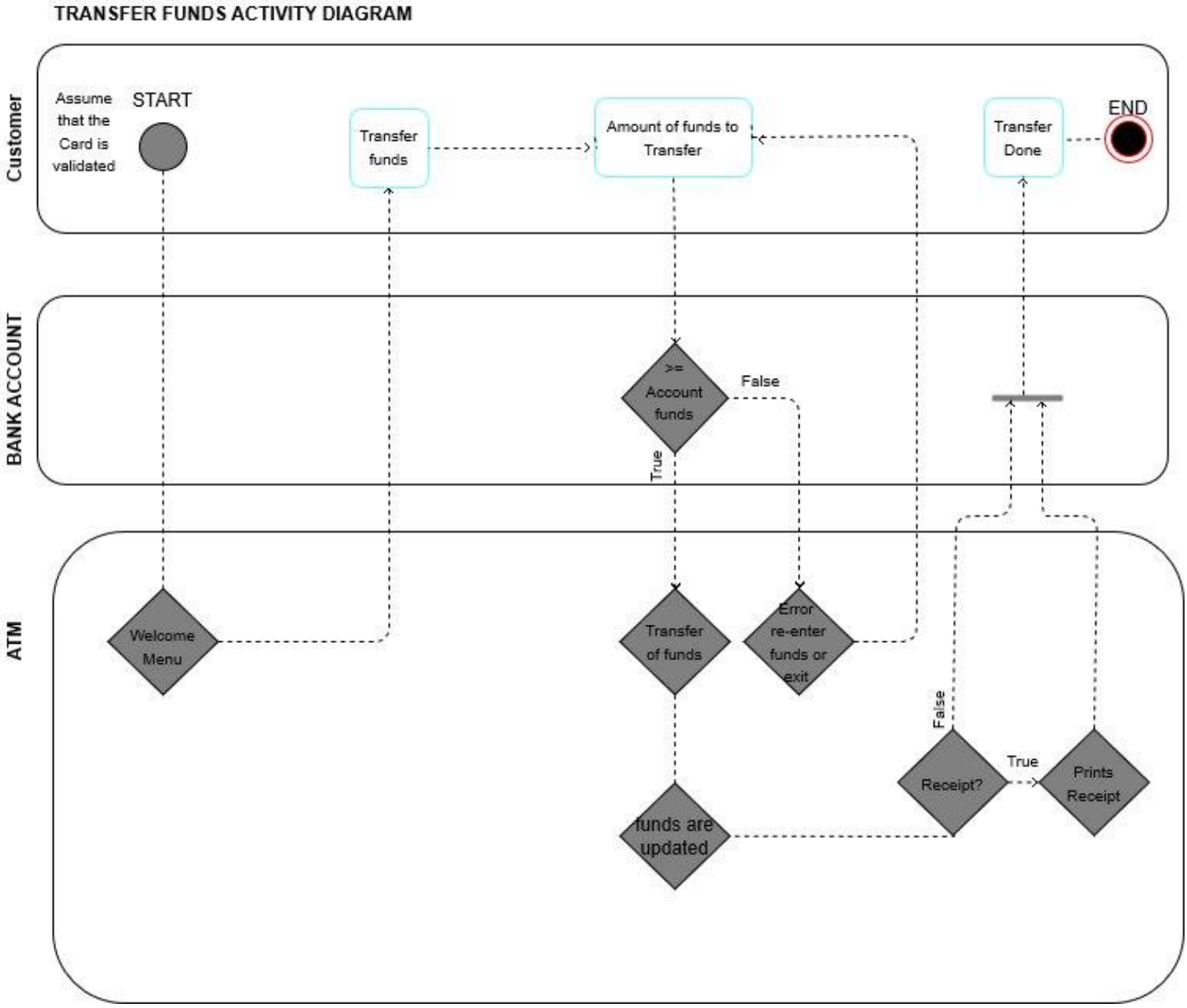
Withdraw Funds	
This class allows user to enter account information and validates the entry. It allows the customer to withdraw money. The requested cash will be dispensed, and receipt will be printed.	check balance ATM

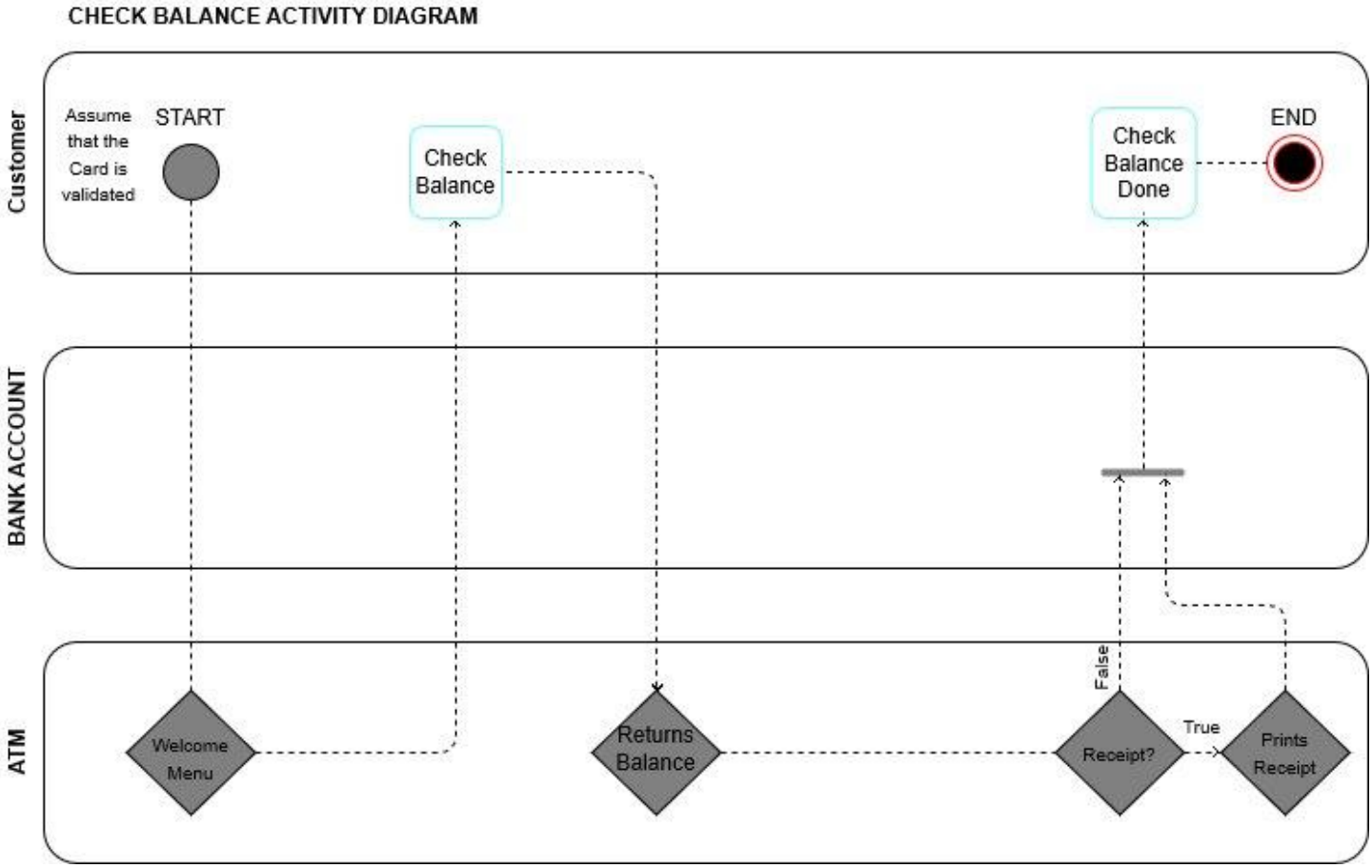
Activity Diagram

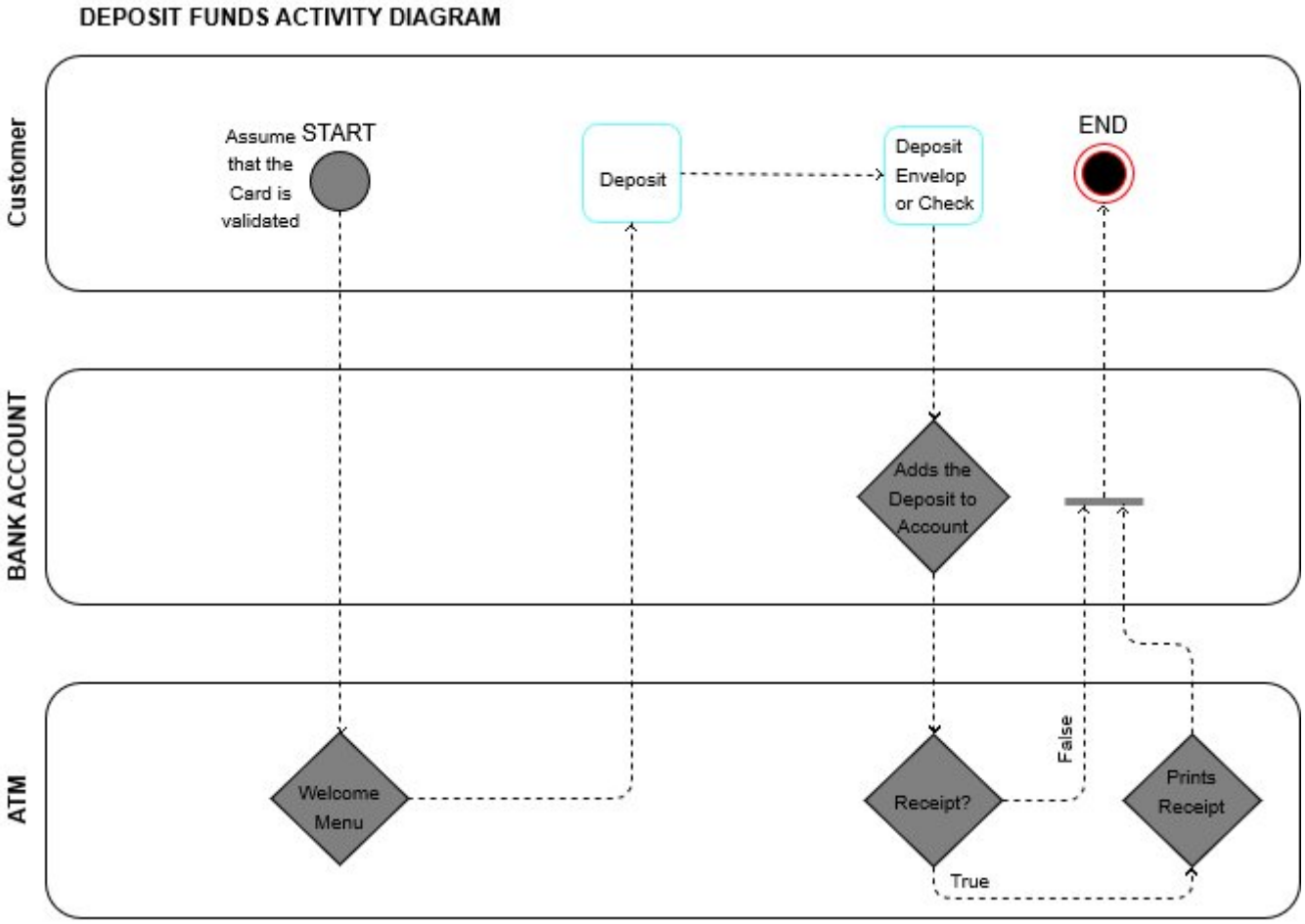
Date of Issuance: May 1, 2020

WITHDRAW



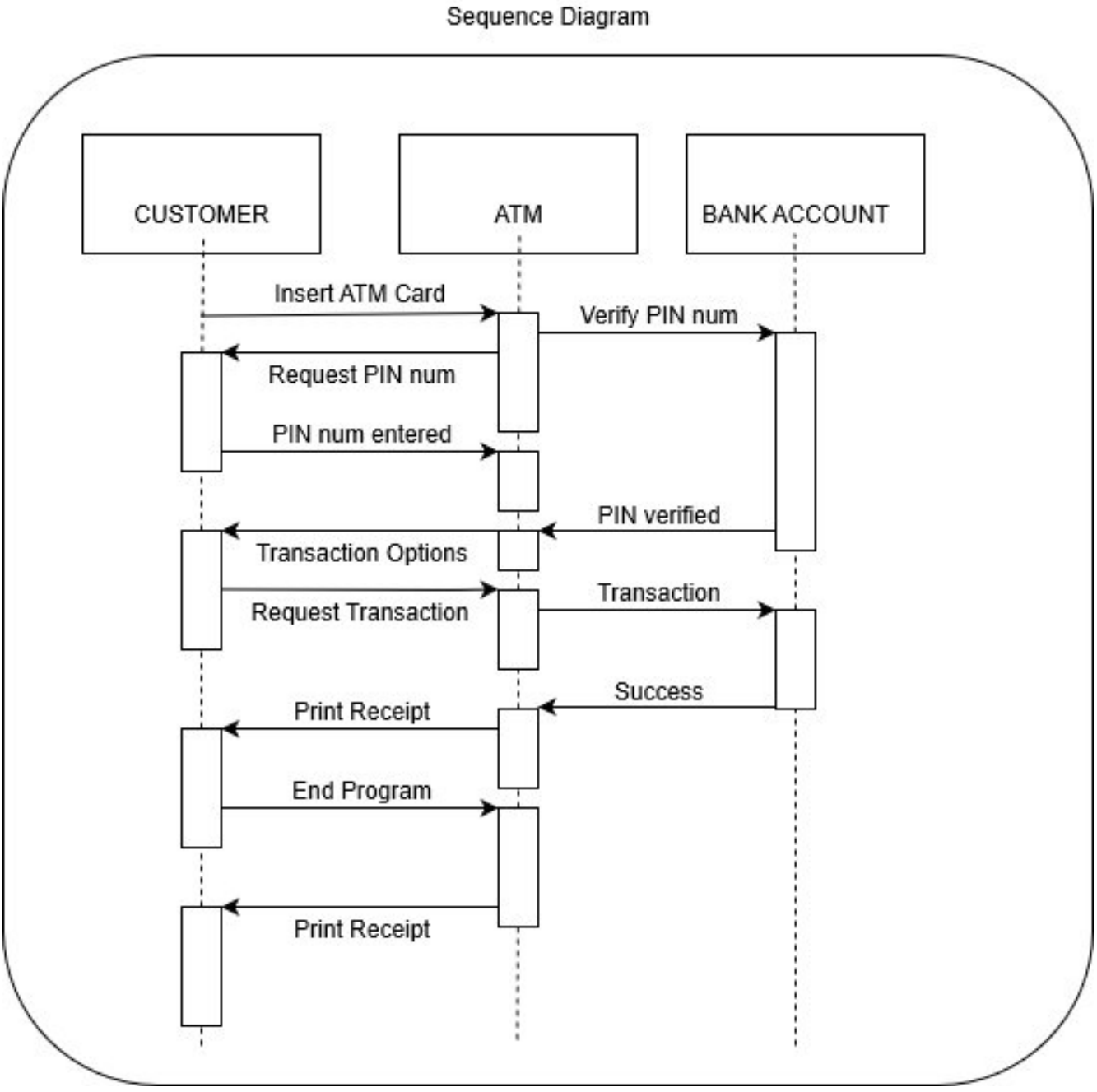

TRANSFER FUNDS ACTIVITY DIAGRAM
Customer
BANK ACCOUNT
ATM
Assume that the Card is validated
START
Transfer funds
Amount of funds to Transfer
Transfer Done
END
>= Account funds
False
True
Welcome Menu
Transfer of funds
Error re-enter funds or exit
False
Receipt?
True
Prints Receipt
funds are updated

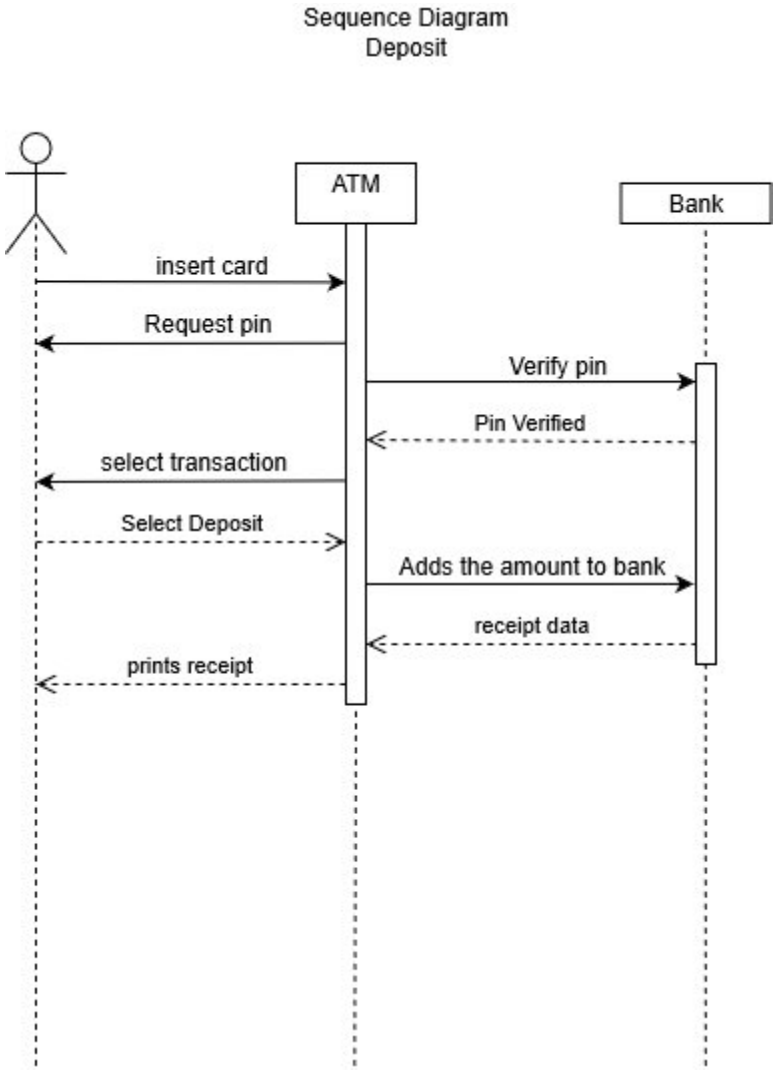

CHECK BALANCE ACTIVITY DIAGRAM
Customer
BANK ACCOUNT
ATM
Assume that the Card is validated
START
Check Balance
Check Balance Done
END
Welcome Menu
Returns Balance
False
Receipt?
True
Prints Receipt

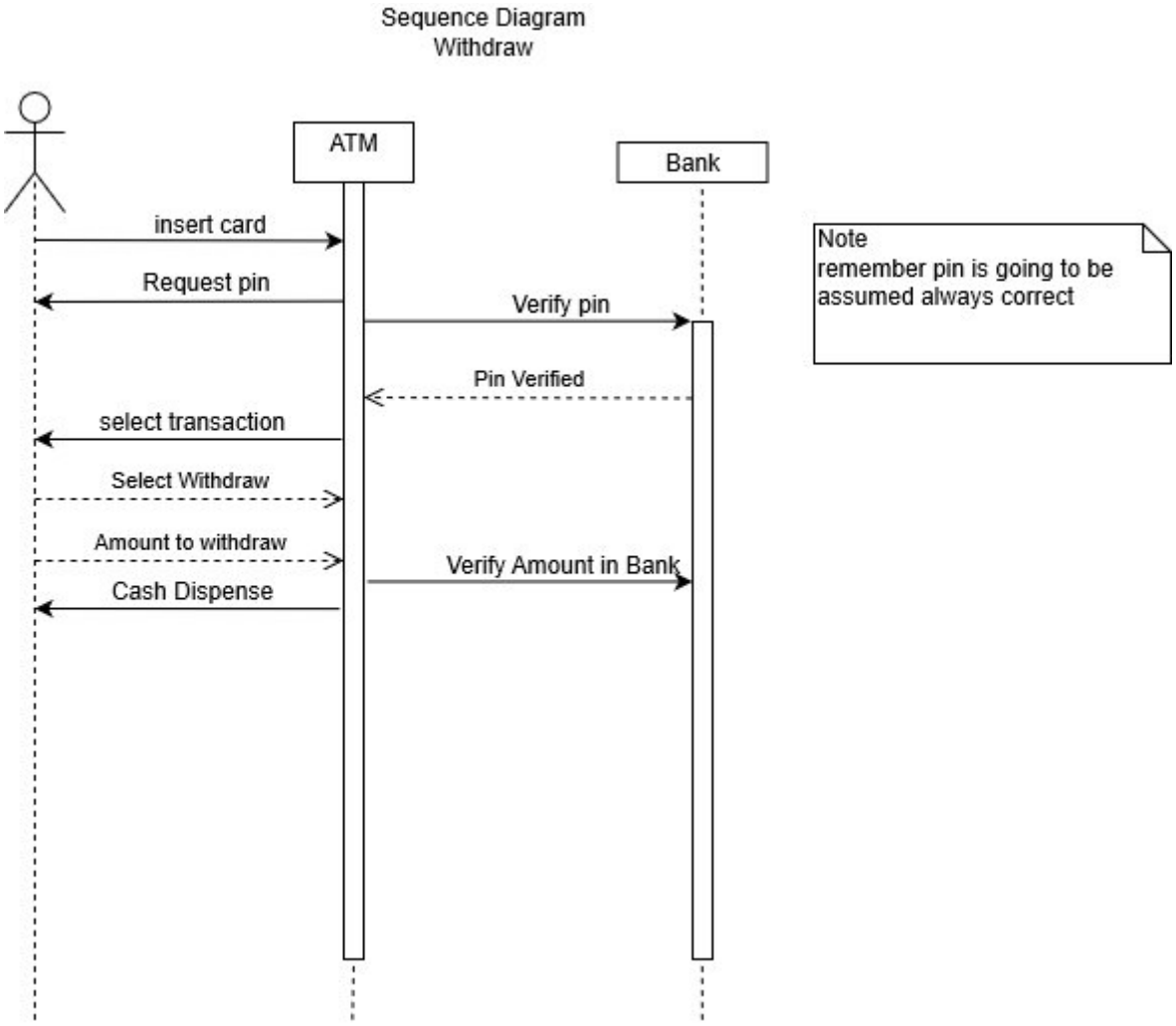

DEPOSIT FUNDS ACTIVITY DIAGRAM
Customer
BANK ACCOUNT
ATM
Assume that the Card is validated
START
Deposit
Deposit Envelop or Check
END
Adds the Deposit to Account
Welcome Menu
Receipt?
False
Prints Receipt
True

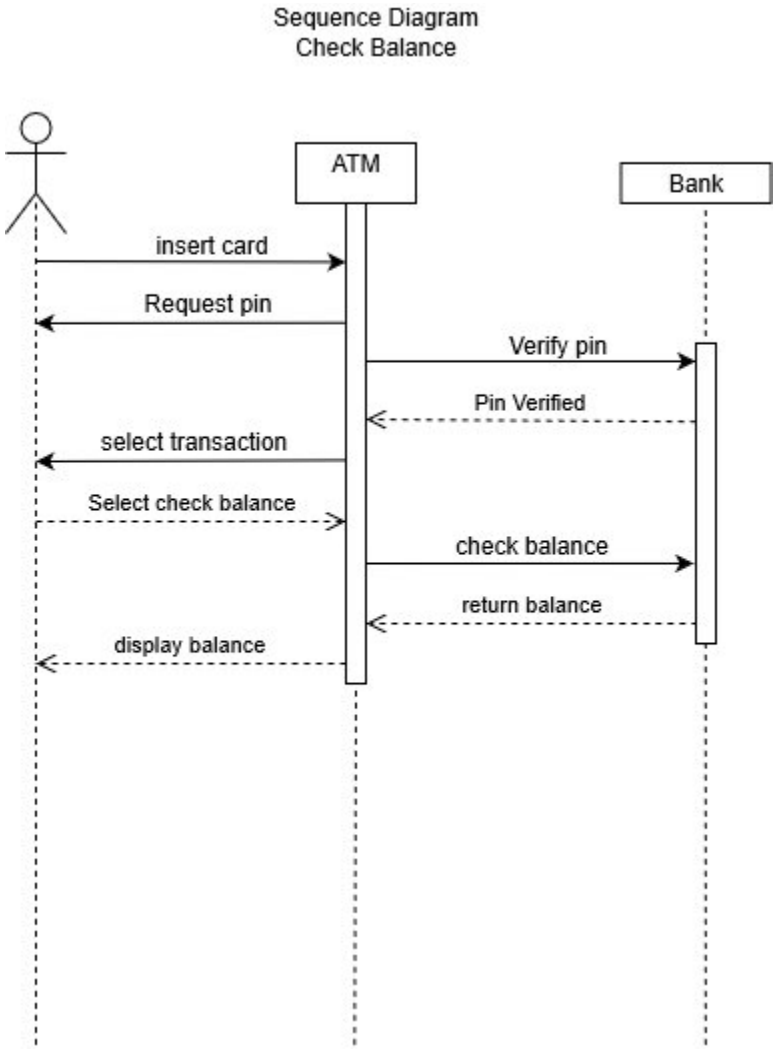
Sequence Diagram

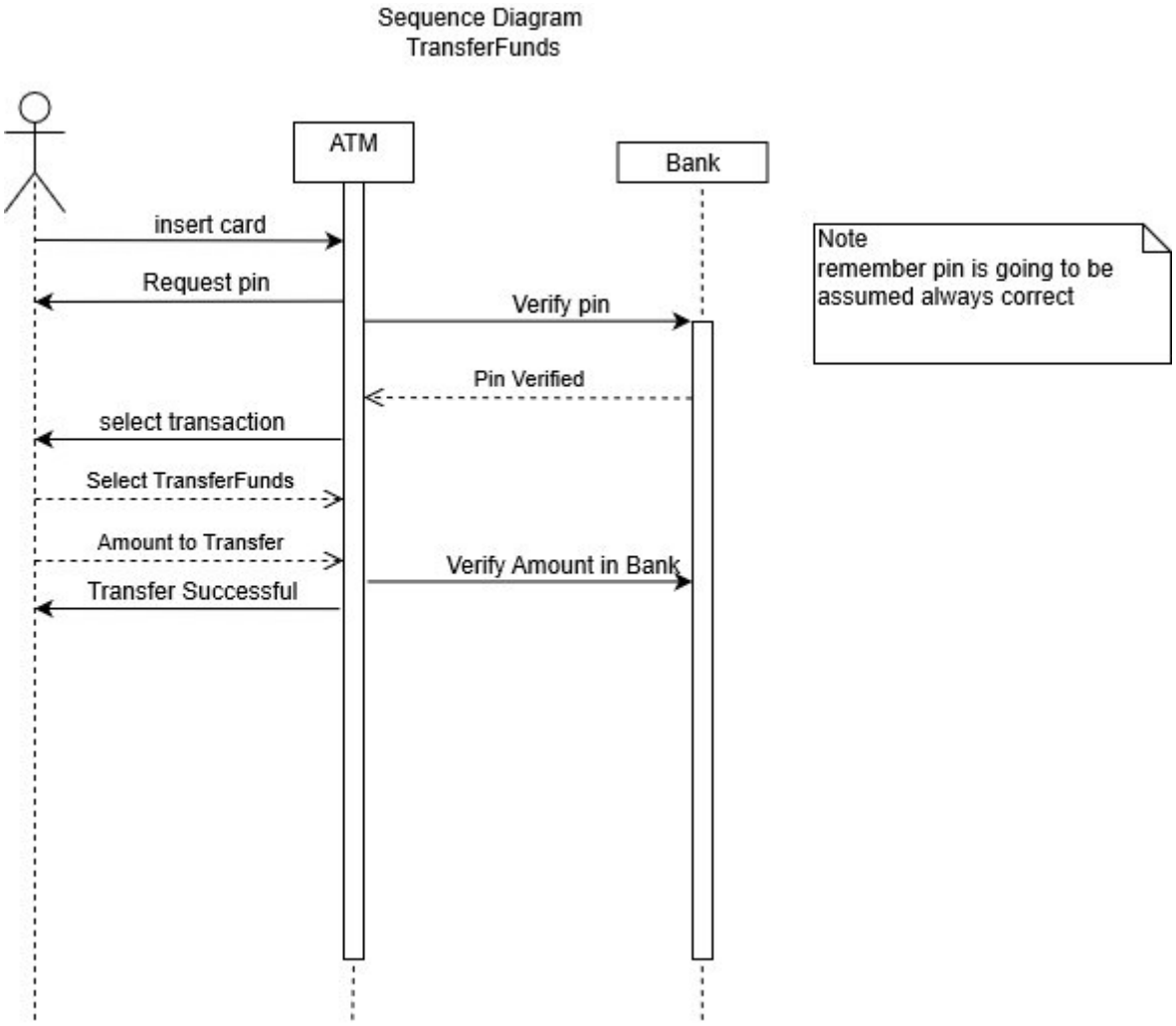
Date of Issuance: April 13, 2020




Sequence Diagram
Deposit
ATM
Bank
insert card
Request pin
Verify pin
Pin Verified
select transaction
Select Deposit
Adds the amount to bank
receipt data
prints receipt

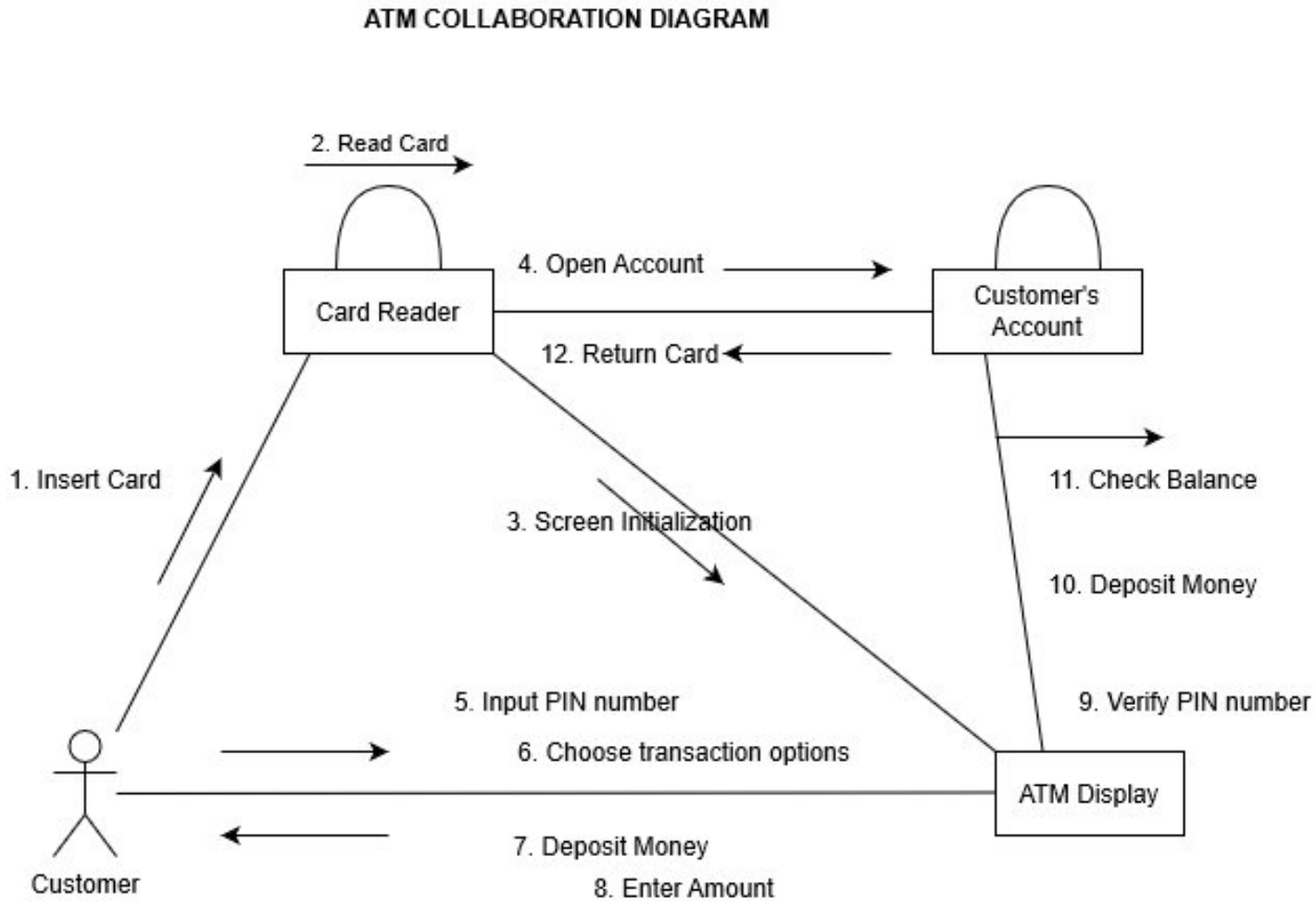

Sequence Diagram
Withdraw
ATM
Bank
insert card
Request pin
Verify pin
Pin Verified
select transaction
Select Withdraw
Amount to withdraw
Verify Amount in Bank
Cash Dispense
Note
remember pin is going to be assumed always correct


Sequence Diagram
Check Balance
ATM
Bank
insert card
Request pin
Verify pin
Pin Verified
select transaction
Select check balance
check balance
return balance
display balance


Sequence Diagram
TransferFunds
ATM
Bank
insert card
Request pin
Verify pin
Pin Verified
select transaction
Select TransferFunds
Amount to Transfer
Verify Amount in Bank
Transfer Successful
Note
remember pin is going to be assumed always correct

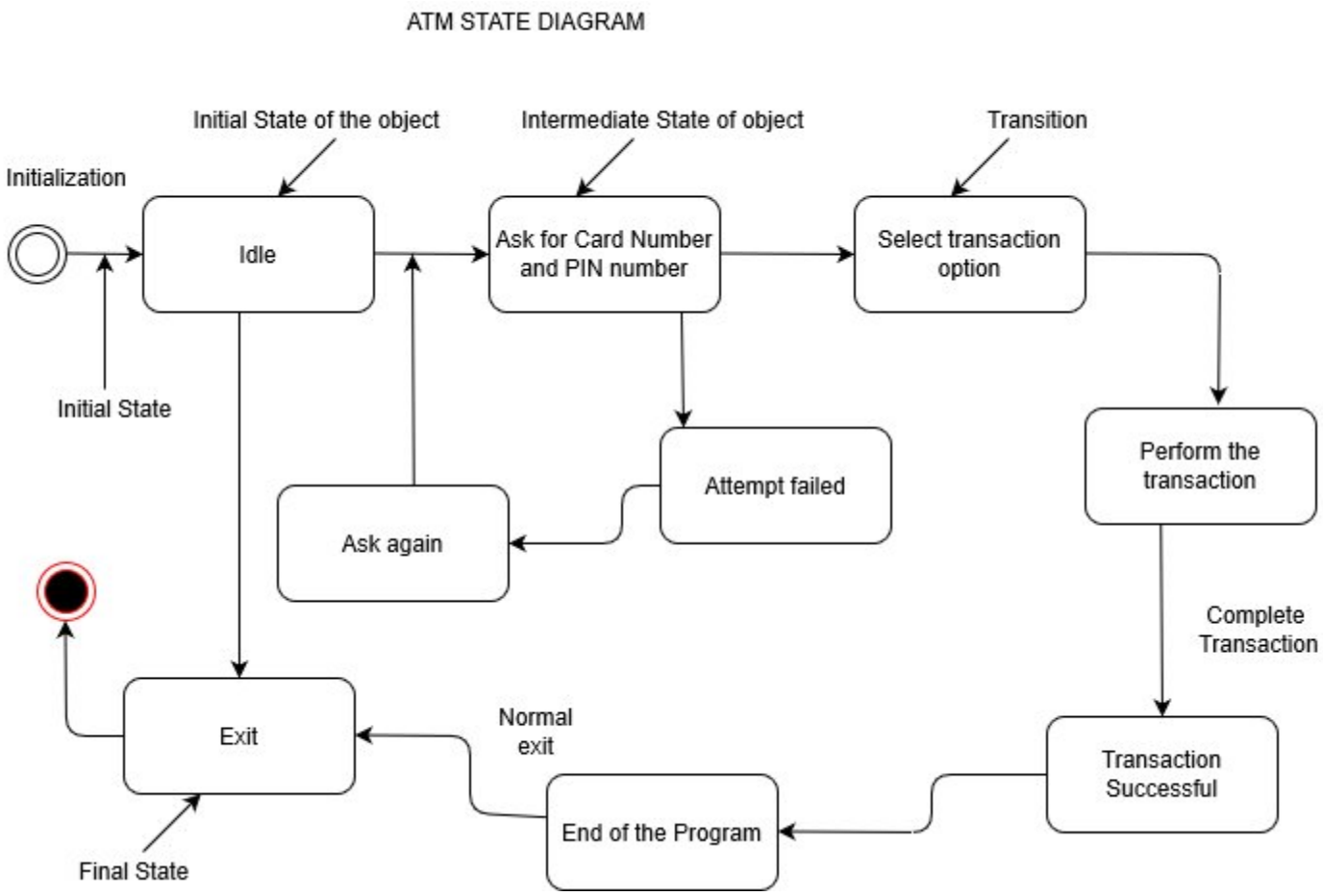
Collaboration Diagram

Date of Issuance: May 3, 2020



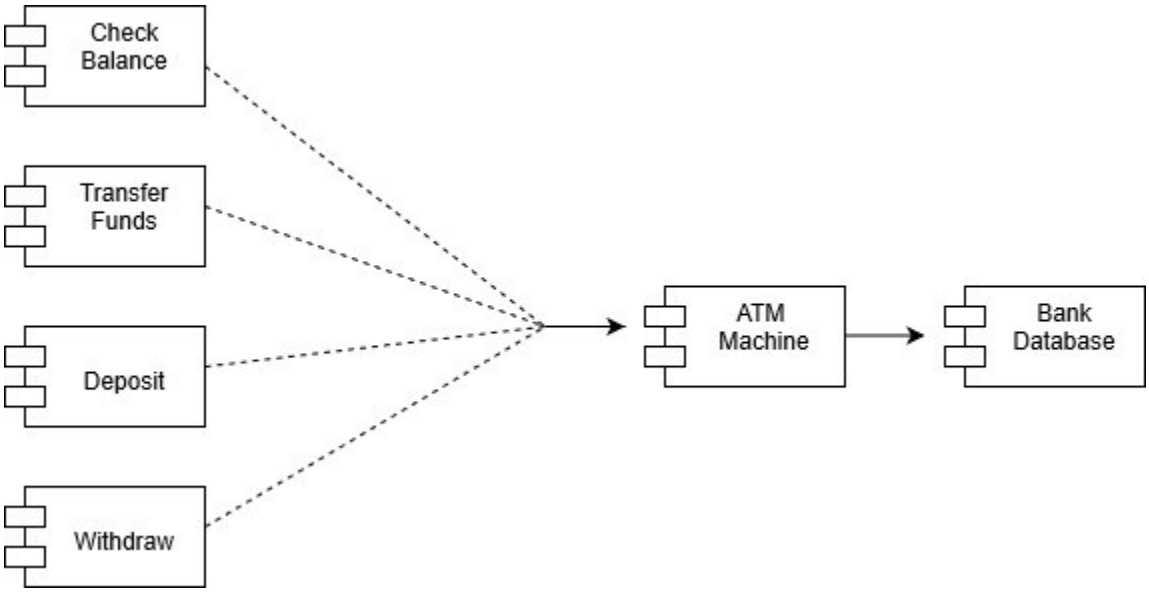
State Diagram

Date of Issuance: May 2, 2020



Component Design

Date of Issuance: May 4, 2020



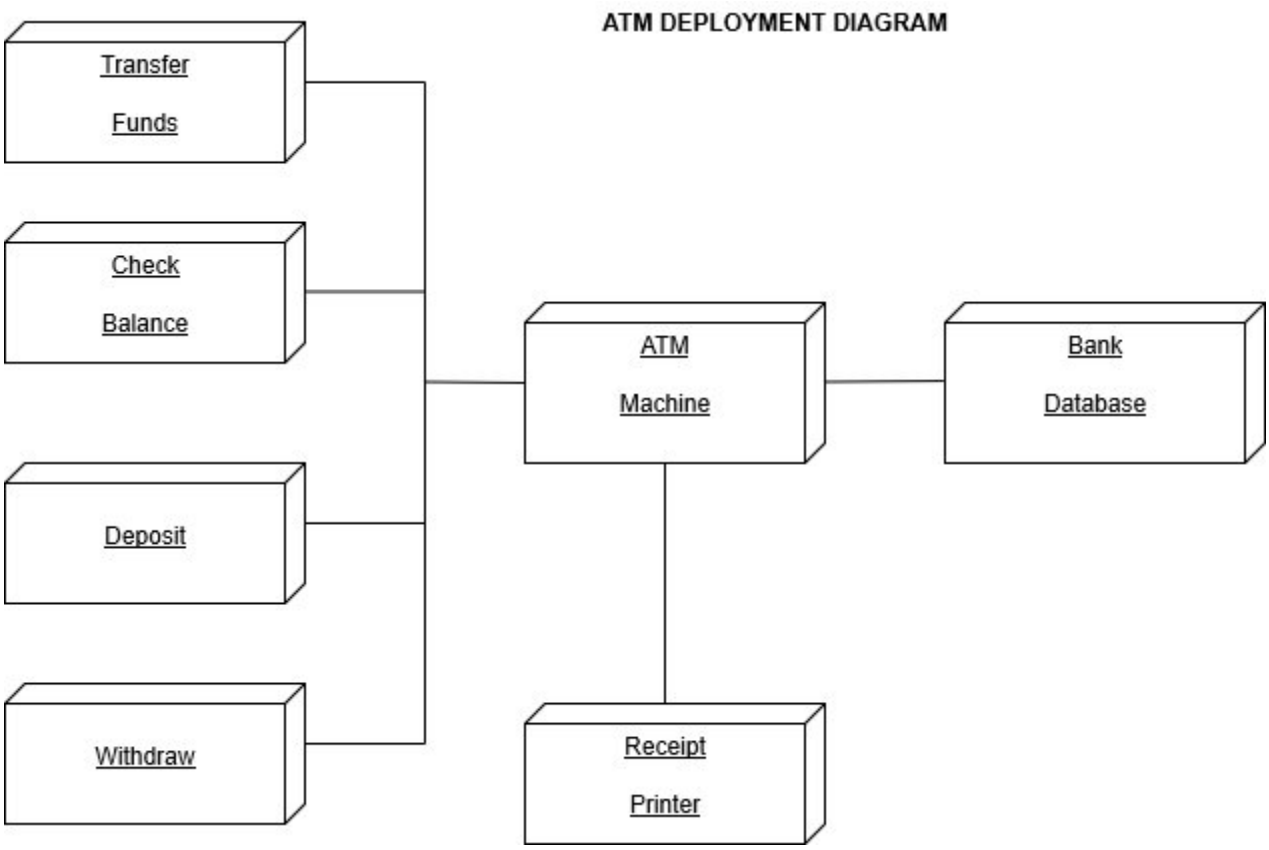
User Interface Design

Date of Issuance: May 5, 2020

```
================================================================================================
================================================================================================
888       888          888                                                    888
888   o   888          888                                                    888
888  d8b  888          888                                                    888
888 d888b 888  .d88b.  888  .d8888b  .d88b.  88888b.d88b.   .d88b.        888888  .d88b.
888d88888b888 d8P  Y8b 888 d88P"    d88""88b 888 "888 "88b d8P  Y8b       888    d88""88b
88888P Y88888 88888888 888 888      888  888 888  888  888 88888888       888    888  888
8888P   Y8888 Y8b.     888 Y88b.    Y88..88P 888  888  888 Y8b.           Y88b.  Y88..88P
888P     Y888  "Y8888  888  "Y8888P  "Y88P"  888  888  888  "Y8888         "Y888  "Y88P"

   8888888b.   .d8888b.     888b    888             d8888 88888888888 888b     d888
   888   Y88b d88P  "88b    8888b   888            d88888     888     8888b   d8888
   888    888 Y88b. d88P    88888b  888           d88P888     888     88888b.d88888
   888   d88P  "Y8888P"     888Y88b 888          d88P 888     888     888Y88888P888
   8888888P"  .d88P88K.d88P 888 Y88b888         d88P  888     888     888 Y888P 888
   888 T88b   888"  Y888P"  888  Y88888        d88P   888     888     888  Y8P  888
   888  T88b  Y88b .d8888b  888   Y8888       d8888888888     888     888   "   888
   888   T88b  "Y8888P" Y88b888    Y888      d88P     888     888     888       888
================================================================================================
================================================================================================



=========================================
Please Enter Card Number: 54321
Please Enter Pin Number : 12345
=========================================


Hello, Welcome to your Bank Account
Choose from the following Transactions or Exit.
1 - Deposit Funds.
2 - Withdraw Funds.
3 - Check Balance.
4 - Transfer Balance.
5 - Exit.
```

Deployment Diagram

Date of Issuance: May 3, 2020



Coding

Date of Issuance: May 6, 2020

ATM.java

```
/**
 * Description: The main runner for the ATM program.
 * CECS 343 Spring 2020
 * ATM Machine: ATM
 * @author Nathaniel Monte De Ramos and Rifa Safeer Shah
 * Date: 05/06/2020
 */

package atmMachine;

import java.util.List;
import java.util.Scanner;
import java.io.FileWriter;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

public class ATM {

	public static void main(String args[]) {

		/* Holder of the card number */
		int cardNumberFromFile = 0;

		/* Holder of the pin number */
		int pinNumberFromFile = 0;

		/* Holder of the balance that is on the file. String so that it can read */
		String balanceFromFile = null;

		Scanner sc = new Scanner(System.in);

		List<String> lines = null;


	System.out.println("========================================================================
==================");


	System.out.println("========================================================================
==================");

		System.out.println(

				"888       888          888
888                    \r\n" +

				"888   o   888          888
888                    \r\n" +
```

```
                        "888  d8b  888          888
888                     \r\n" +

                        "888 d888b 888  .d88b.  888  .d8888b  .d88b.  88888b.d88b.   .d88b.
888888  .d88b.          \r\n" +

                        "888d88888b888 d8P  Y8b 888 d88P\"    d88\"\"88b 888 \"888 \"88b d8P
Y8b       888    d88\"\"88b          \r\n" +

                        "88888P Y88888 88888888 888 888      888  888 888  888  888 88888888
888    888  888          \r\n" +

                        "8888P   Y8888 Y8b.     888 Y88b.    Y88..88P 888  888  888 Y8b.
Y88b.  Y88..88P          \r\n" +

                        "888P     Y888  \"Y8888  888  \"Y8888P  \"Y88P\"  888  888  888  \"Y8888
\"Y888  \"Y88P\"           \r\n" +

                        "
\r\n" +

                        "   8888888b.   .d8888b.     888b    888              d8888 88888888888
888b     d888 \r\n" +

                        "   888   Y88b d88P  \"88b    8888b   888             d88888     888
8888b   d8888 \r\n" +

                        "   888    888 Y88b. d88P    88888b  888            d88P888     888
88888b.d88888 \r\n" +

                        "   888   d88P  \"Y8888P\"     888Y88b 888           d88P 888     888
888Y88888P888 \r\n" +

                        "   8888888P\"  .d88P88K.d88P 888 Y88b888          d88P  888     888
888 Y888P 888 \r\n" +

                        "   888 T88b   888\"  Y888P\"  888  Y88888         d88P   888     888
888  Y8P  888 \r\n" +

                        "   888  T88b  Y88b .d8888b  888   Y8888        d8888888888     888
888   \"   888 \r\n" +

                        "   888   T88b  \"Y8888P\" Y88b888    Y888       d88P     888     888
888       888");


      System.out.println("================================================================================
==================");


      System.out.println("================================================================================
==================\n\n\n");

            System.out.println("==========================================");

            /* Give a 5 digit card number */
            System.out.print("Please Enter Card Number: ");

            /* Takes the card number from the user */
```

```
String cardNum1 = sc.next();

boolean willUserExit = false;

try {

	/* Search for card number text file */
	Path fileScan = Paths.get(cardNum1 + ".txt");

	/* Put all lines in a list of strings */
	lines = Files.readAllLines(fileScan);

	/* The card number that is on the file (file name) */
	cardNumberFromFile = Integer.parseInt(lines.get(0));

	/* the pin number that is on the file */
	pinNumberFromFile = Integer.parseInt(lines.get(1));

	/* Balance amount on the file */
	balanceFromFile = lines.get(2);
} // End of try block

/**
 * Expected in the file:
 * line 0 -> card number
 * line 1 -> pin number
 * line 2 -> checking balance
 */
catch(Exception e) {

	System.out.println("Error: Card does not Exist");

	System.exit(0);
} // End of catch block

/**
 * There are three attempts to enter pin number.
 */
for(int i = 0; i < 3; i++) {

	/* 5 digit pin number */
	System.out.print("Please Enter Pin Number : ");

	/**
	 * If card number and pin number are right then this menu will be shown
	 * If not then the program will exit after 3 attempts
	 */
	try {

		String pinNum = sc.next();

		if(Integer.parseInt(pinNum) == pinNumberFromFile) {

			i += 2;

			Account userBankAccount = new Account(cardNumberFromFile,
pinNumberFromFile, balanceFromFile);//THE ACCOUNT OF THE CURRENT USER

			/* Welcome Message */
```

```
	System.out.println("=========================================\n\n");

                                System.out.println("Hello, Welcome to your Bank Account");

                                /**
                                 * After the program shows the balance this menu is shown.
                                 */
                                while(!willUserExit) {

                                        menuChoice();

                                        /* return to menu */
                                        System.out.println("Would like to return to MENU?");

                                        System.out.println("1. Yes\n2. No");

                                        int userReturn = sc.nextInt();

                                        if(userReturn == 1) {

                                                willUserExit = false;
                                        } // End of if-statement

                                        else if(userReturn == 2) {

                                                willUserExit = true;
                                        } // End of else if-statement
                                } // End of while loop

                                /* Print the receipt */
                                PrintReceipt.printReceipt();
                        } // End of if-statement

                        /**
                         * Else if the pin number entered by the user is right.
                         */
                        else {

                                System.out.println("Wrong Pin Number. Please try again");

                                /**
                                 * If the user has used the 3 attempts. Else go back to loop.
                                 */
                                if(i == 2) {

                                        System.out.println("Login Attempt Exceeded Allowed
Ammount.");

                                        System.exit(0);
                                } // End of if-statement
                        } // End of else-statement
                } // End of try block

                catch(NumberFormatException ex) {

                        System.out.println("Error: Wrong Pin Number. Please try again");
                } // End of catch block
        } // End of for loop
```

```
		/**
		 * Save to file.
		 * Opens the same text then writes the new balance.
		 */
		try {

			/* String value of card number */
			String numCard = String.valueOf(Account.cardNum);

			/* String value of pin number */
			String numPin = String.valueOf(Account.pinNum);

			/* String value of balance */
			String numBalance = String.valueOf(Account.balance);

			FileWriter fw = new FileWriter(numCard + ".txt");

			fw.write(numCard + "\n" + numPin + "\n" + numBalance);

			fw.close();
		} // End of try block

		catch(Exception e) {

			System.out.print("Writing to file problem");
		} // End of catch block
	} // End of main

	public static void menuChoice() {

		Scanner sc = new Scanner(System.in);

		System.out.println("Choose from the following Transactions or Exit.");

		System.out.println("1 - Deposit Funds.");

		System.out.println("2 - Withdraw Funds.");

		System.out.println("3 - Check Balance.");

		System.out.println("4 - Transfer Balance.");

		System.out.println("5 - Exit.");

		/* Choice for menu */
		int choice = sc.nextInt();

		/**
		 * Switch case to send to different class.
		 */
		switch(choice) {

			case 1:

				Deposit.deposit();

				break;
```

```
                    case 2:
                           Withdraw.withdraw();
                           break;
                    case 3:
                           CheckBalance.checkBalance();
                           break;
                    case 4:
                           TransferFund.transferFund();
                           break;
                    case 5:
                           System.out.println("Thank you for using our ATM. Have a nice day.");
                           System.exit(0);
             } // End of switch case
      } // End of menuChoice
} // End of ATM class
```

Account.java

```
/**
 * Description: The account operator for the ATM machine.
 * CECS 343 Spring 2020
 * ATM Machine: Account
 * @author Nathaniel Monte De Ramos and Rifa Safeer Shah
 * Date: 05/06/2020
 */

package atmMachine;

public class Account {

      /* Card number for the account */
      static int cardNum;

      /* Pin number for the account */
      static int pinNum;

      /* Balance in the account */
      static String balance;

      public Account(int card,int pin, String bal) {

            Account.cardNum = card;

            Account.pinNum = pin;

            Account.balance = bal;
```

```
	} // End of Account

	/**
	 * Gets the balance in the account.
	 * @return balance. Balance as a double.
	 */
	public static double getBalance() {

		return Double.parseDouble(balance);
	} // End of getBalance

	/**
	 * Gets the card number.
	 * @return card number. Card Number as an integer.
	 */
	public static int getCardNum() {

		return cardNum;
	} // End of getCardNum

	/**
	 * Sets the balance in the account.
	 * @param bal balance in the account.
	 */
	public static void setBalance(double bal) {

		if (Double.parseDouble(balance) >= 0) {

			balance = String.valueOf(bal);
		} // End of if-statement
	} // End of setBalance
} // End of Account class
```

CheckBalance.java

```
/**
 * Description: The class that checks balance in the account.
 * CECS 343 Spring 2020
 * ATM Machine: CheckBalance
 * @author Nathaniel Monte De Ramos and Rifa Safeer Shah
 * Date: 05/06/2020
 */

package atmMachine;

/**
 * Checks the balance in the account.
 */
public class CheckBalance {

	public static void checkBalance() {

		System.out.println("===============CHECK BALANCE===============");

		System.out.println("Card Number          : " + Account.getCardNum());

		System.out.println("Checking Balance     : " + Account.getBalance());
```

```
            System.out.println("=========================================");
      } // End of checkBalance
} // End of CheckBalance class
```

Deposit.java

```
/**
 * Description: The class that allows customers to deposit money into the account.
 * CECS 343 Spring 2020
 * ATM Machine: Deposit
 * @author Nathaniel Monte De Ramos and Rifa Safeer Shah
 * Date: 05/06/2020
 */

package atmMachine;

import java.util.Scanner;

/**
 * Adds amount to the account.
 */
public class Deposit {

      public static void deposit() {

            /* Initial for deposit amount */
            double depositAmmount;

            /* Initial for the account balance */
            double accBalance;

            System.out.println("==============DEPOSIT==============");

            /* For scanning the user input for deposit */
            Scanner scan = new Scanner(System.in);

            /* Show the user the balance before deposit */
            System.out.println("Your balance is $" + Account.getBalance());

            System.out.println("How much would you like to deposit?");

            /* Taking in the deposit */
            depositAmmount = scan.nextDouble();

            /* The new balance */
            accBalance = Account.getBalance() + depositAmmount;

            /* Print out the new balance */
            System.out.println("Your new balance is $" + accBalance);

            /* Set the customer balance to the new balance */
            Account.setBalance(accBalance);

            System.out.println("====================================");
      } // End of deposit
} // End of Deposit class
```

PrintReceipt.java

```
/**
 * Description: The class handles the printing of the receipts.
 * CECS 343 Spring 2020
 * ATM Machine: PrintReceipt
 * @author Nathaniel Monte De Ramos and Rifa Safeer Shah
 * Date: 05/06/2020
 */

package atmMachine;

import java.util.Scanner;

public class PrintReceipt {

	public static void printReceipt() {

		Scanner scan = new Scanner(System.in);

		System.out.println("Would you like a Receipt?\n1. Yes\n2. No");

		int choice = scan.nextInt();

		if(choice == 1) {

			System.out.println("====================ATM====================");

			System.out.println("Location: LONG-BEACH-CA");

			System.out.println("ATM #: 16542M");

			System.out.println("\nUser Card: " + Account.getCardNum());

			System.out.println("Transaction #:  5623");

			System.out.println("Total Balance: " + Account.getBalance());

			System.out.println("Thank you for using our ATM.");

			System.out.println("For questions, call RIFA\nBusiness customers call
NATHAN");

			System.out.println("============================================");
		} // End of if-statement

		else {

			System.out.println("Thank you for using our ATM. Have a nice day.");
		} // End of else-statement
	} // End of printReceipt
} // End of PrintReceipt
```

TransferFunds.java

```java
/**
 * Description: The class handles the transfer transaction for the accounts.
 * CECS 343 Spring 2020
 * ATM Machine: TransferFunds
 * @author Nathaniel Monte De Ramos and Rifa Safeer Shah
 * Date: 05/06/2020
 */

package atmMachine;

import java.io.FileWriter;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.List;
import java.util.Scanner;

/**
 * Transfers fund from open file to another.
 */
public class TransferFund {

	public static void transferFund() {

		/* Amount to be transferred */
		double transferAmount;

		Scanner scan = new Scanner(System.in);

		System.out.println("==================TRANSFER===============");

		/* Balance from the user */
		System.out.println("Your balance is " + Account.getBalance());

		/**
		 * User gets 3 tries.
		 */
		for(int i = 0; i < 3; i++) {

			System.out.println("How much would you like to Transfer?");

			transferAmount = scan.nextDouble();

			if(transferAmount >= Account.getBalance()) {

				System.out.println("Transfer Amount is Greater than your
Balance.\nPlease Try Again.");
			} // End of if-statement

			else {

				/* For the user that is transferring */
				double newBalance = Double.parseDouble(Account.balance) -
transferAmount;

				System.out.println("Enter the Card Number of the Person you want to
transfer the money ");

				/* Account that the money is being transferred to */
```

```
                        int cardNum2 = scan.nextInt();

                        try {

                                /* Find the file with the card number */
                                Path fileScan = Paths.get(cardNum2 + ".txt");

                                /* Make the list */
                                List<String> lines = Files.readAllLines(fileScan);

                                String numCard2 = lines.get(0);

                                String numPin2 = lines.get(1);

                                /* Take the third number. Account balance of the transfee */
                                String balanceFromFile2 = lines.get(2);

                                /* Add the transfer amount */
                                double accBalance2 = Double.parseDouble(balanceFromFile2) +
transferAmount;


                                FileWriter fw = new FileWriter(numCard2 + ".txt");

                                fw.write(numCard2 + "\n" + numPin2 + "\n" + accBalance2);

                                fw.close();

                                System.out.println("Transfer was successful");
                        } // End of try block

                        catch(Exception e) {

                                System.out.println("Card does not Exist");

                                System.exit(0);
                        } // End of catch block

                        Account.setBalance(newBalance);

                        System.out.println("========================================");

                        i += 2;
                    } // End of else-statement
                } // End of for loop
        } // End of transferFund
} // End of TransferFund
```

Withdraw.java

```
/**
 * Description: This class allows users to take money out of the account.
 * CECS 343 Spring 2020
 * ATM Machine: Withdraw
 * @author Nathaniel Monte De Ramos and Rifa Safeer Shah
 * Date: 05/06/2020
 */
```

```java
package atmMachine;

import java.util.Scanner;

/**
 * Subtracts an amount from the account balance.
 */
public class Withdraw {

	public static void withdraw() {

		double withdrawAmount;

		double accBalance;

		Scanner scan = new Scanner(System.in);

		System.out.println("===============WITHDRAW===============");

		System.out.println("Your balance is " + Account.getBalance());

		/**
		 * User gets 3 tries.
		 */
		for(int i = 0; i < 3; i++) {

			System.out.println("How much would you like to withdraw?");

			withdrawAmount = scan.nextDouble();

			if (withdrawAmount >= Account.getBalance()) {

				System.out.println("Withdrawal Amount is Greater than your
Balance.\nPlease Try Again.");
			} // End of if-statement

			else {

				accBalance = Account.getBalance() - withdrawAmount;

				System.out.println("Your new balance is " + accBalance);

				Account.setBalance(accBalance);

				i += 2;
			} // End of else-statement
		} // End of for loop

		System.out.println("=====================================");
	} // End of withdraw
} // End of Withdraw class
```

0.txt

```
0
0
null
```

54321.txt

```
54321
12345
2000
```

98765.txt

```
98765
56789
2500.0
```

Unit Testing

Date of Issuance: May 7, 2020

Unit Testing 1: Entering Card Number

Test 1: Card number is valid.

Program will ask for a PIN number.

Test 2: Card number is invalid, or card number can't be found.

Program will show an error message then exit.

Test 3: User entered a card number character that's not an integer.

Program will show an error message then exit.

Test 4: User pressed enter without entering a card number.

Program will go to the next line and wait until user puts anything.

Unit Testing 2: Entering PIN Number

Test 1: PIN number is valid.

Program will take the user to the menu.

Test 2: PIN number is invalid.

Program will give the user 2 more tries.

After the third try the program will exit.

Test 3: User entered a PIN number character that's not an integer.

Program will see the entered PIN as invalid.

Test 4: User pressed enter without entering a card number.

Program will go to the next line and wait until user puts anything.

Runtime Output

Date of Issuance: May 6, 2020

```
Hello, Welcome to your Bank Account
Choose from the following Transactions or Exit.
1 - Deposit Funds.
2 - Withdraw Funds.
3 - Check Balance.
4 - Transfer Balance.
5 - Exit.
1
==============DEPOSIT==============
Your balance is $2000.0
How much would you like to deposit?
1000
Your new balance is $3000.0
====================================
Would like to return to MENU?
1. Yes
2. No
1
Choose from the following Transactions or Exit.
1 - Deposit Funds.
2 - Withdraw Funds.
3 - Check Balance.
4 - Transfer Balance.
5 - Exit.
2
==============WITHDRAW==============
Your balance is 3000.0
How much would you like to withdraw?
500
Your new balance is 2500.0
====================================
Would like to return to MENU?
1. Yes
2. No
1
Choose from the following Transactions or Exit.
1 - Deposit Funds.
2 - Withdraw Funds.
3 - Check Balance.
```

```
3 - Check Balance.
4 - Transfer Balance.
5 - Exit.
3
==============CHECK BALANCE==============
Card Number          : 54321
Checking Balance     : 2500.0
=========================================
Would like to return to MENU?
1. Yes
2. No
1
Choose from the following Transactions or Exit.
1 - Deposit Funds.
2 - Withdraw Funds.
3 - Check Balance.
4 - Transfer Balance.
5 - Exit.
4
==================TRANSFER================
Your balance is 2500.0
How much would you like to Transfer?
500
Enter the Card Number of the Person you want to transfer the money
98765
Transfer was successful
=========================================
Would like to return to MENU?
1. Yes
2. No
1
Choose from the following Transactions or Exit.
1 - Deposit Funds.
2 - Withdraw Funds.
3 - Check Balance.
4 - Transfer Balance.
5 - Exit.
5
Thank you for using our ATM. Have a nice day.
```

Summary

Date of Issuance: May 7, 2020

Project:

ATM desktop software

Function:

1. Withdraw money from the account.
2. Transfer money from one account to the other account.
3. Check the balance in an account.
4. Deposit money to the account.

Report:

1. Problem Statement
2. Use Case and Scenarios
3. Domain Analysis
4. Requirement Modeling
    a. Class Diagram
    b. Detailed Class Diagram
    c. CRC
    d. Activity Diagram
    e. Sequence Diagram
    f. Collaboration Diagram
    g. State Diagram
5. Component Design
6. User Interface Design
7. Deployment Diagram
8. Coding
9. Unit Testing
10. Runtime Output
11. Summary