Programming 261

Lecturer: Mr Raymond Hood

Date: 10 April 2023

Group Members:

Lehlohonolo Maleka, 577509

Nicolas Buys, 576529

Table of Contents

ATM TRANSACTION APP CONSOLE:

The ATM transaction app allows the user to enter a pin and account number in order to access it. We have created a menu consisting of 5 options. The console app allows the user to view their current balance, enter a withdraw amount and make a cash deposit into their account.

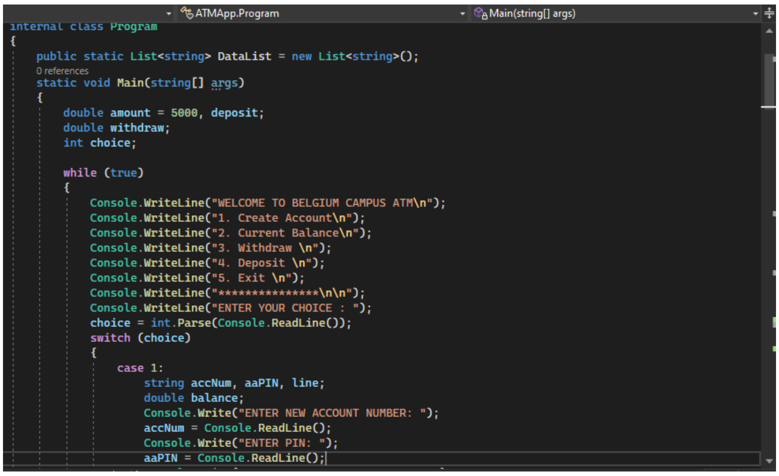
Pseudocode:

```
Declare variables data type + Variables for amount and pin
Initialize account
    double amount, double withdraw, int choice
While (true) loop through application until user exits application
Initialize menu
    Console.Writeline “Welcome to Belgium Campus Atm”
    Console.WriteLine("1. Create Account\n");
    Console.WriteLine("2. Current Balance\n");
    Console.WriteLine("3. Withdraw \n");
    Console.WriteLine("4. Deposit \n");
    Console.WriteLine("5. Exit \n");
    Console.WriteLine("***************\n\n");
    Console.WriteLine("ENTER YOUR CHOICE : ");
    choice = int.Parse(Console.ReadLine());
Initialize account number, balance and pin
Get user input: Search accountNumber and Pin
Display user balance
Initialize withdrawal
if
    withdrawal balance is < than R100
print “enter amount above R100”
else if
    withdrawal is > amount 1000
Console.Writeline (“Sorry Insufficent funds. Minimum is R100”
Declare balance variable
```

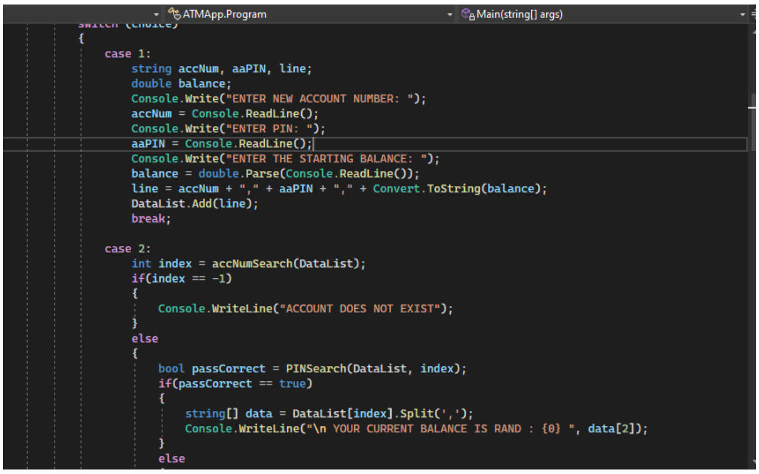
```
        Balance =balance -Withdraw
return and calculate new balance
print new balance
Initialize deposit amount
        Declare variables: Deposit, amount
                Amount= amount+ deposit
Console.Writeline (“you have successfully deposited amount”)
Exit application
Print “Thank you for using “Belgium Campus ATM service”
```

C# Application Code:

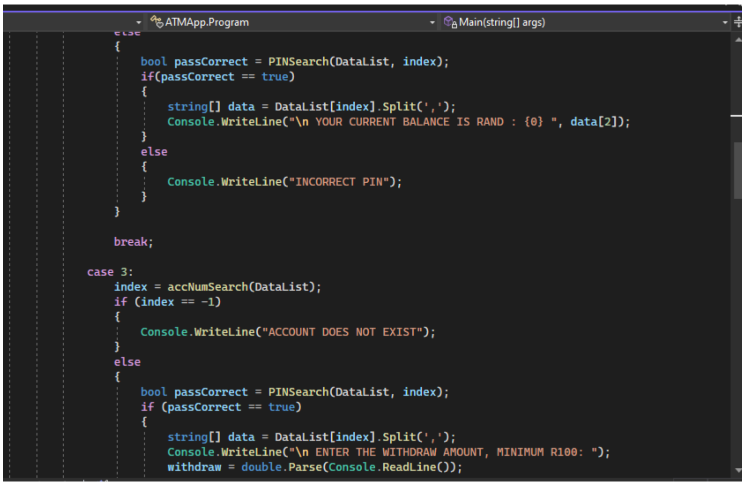
The below image shows the code that was used to write and execute the program.



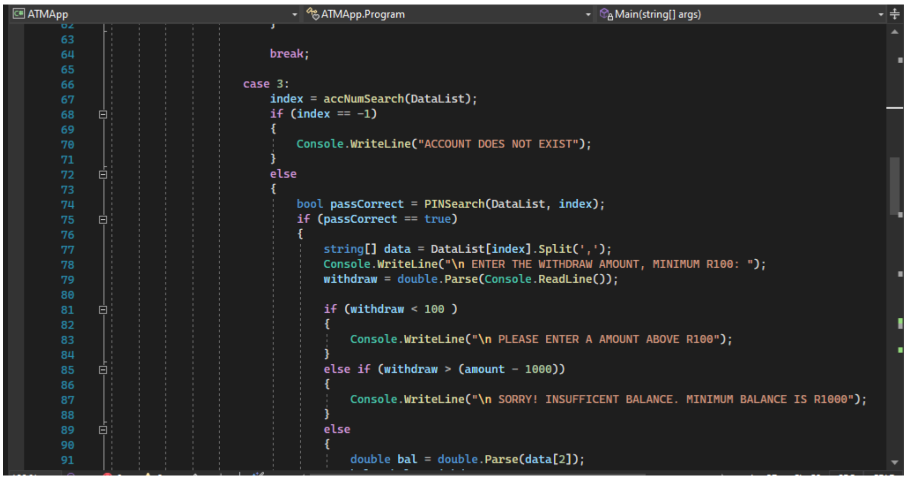
```csharp
internal class Program
{
    public static List<string> DataList = new List<string>();
    static void Main(string[] args)
    {
        double amount = 5000, deposit;
        double withdraw;
        int choice;

        while (true)
        {
            Console.WriteLine("WELCOME TO BELGIUM CAMPUS ATM\n");
            Console.WriteLine("1. Create Account\n");
            Console.WriteLine("2. Current Balance\n");
            Console.WriteLine("3. Withdraw \n");
            Console.WriteLine("4. Deposit \n");
            Console.WriteLine("5. Exit \n");
            Console.WriteLine("***************\n\n");
            Console.WriteLine("ENTER YOUR CHOICE : ");
            choice = int.Parse(Console.ReadLine());
            switch (choice)
            {
                case 1:
                    string accNum, aaPIN, line;
                    double balance;
                    Console.Write("ENTER NEW ACCOUNT NUMBER: ");
                    accNum = Console.ReadLine();
                    Console.Write("ENTER PIN: ");
                    aaPIN = Console.ReadLine();
```



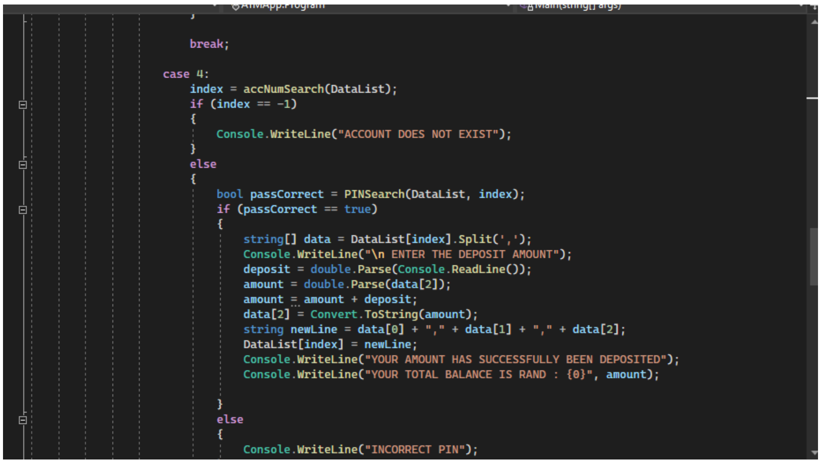
```csharp
            switch (choice)
            {
                case 1:
                    string accNum, aaPIN, line;
                    double balance;
                    Console.Write("ENTER NEW ACCOUNT NUMBER: ");
                    accNum = Console.ReadLine();
                    Console.Write("ENTER PIN: ");
                    aaPIN = Console.ReadLine();
                    Console.Write("ENTER THE STARTING BALANCE: ");
                    balance = double.Parse(Console.ReadLine());
                    line = accNum + "," + aaPIN + "," + Convert.ToString(balance);
                    DataList.Add(line);
                    break;

                case 2:
                    int index = accNumSearch(DataList);
                    if(index == -1)
                    {
                        Console.WriteLine("ACCOUNT DOES NOT EXIST");
                    }
                    else
                    {
                        bool passCorrect = PINSearch(DataList, index);
                        if(passCorrect == true)
                        {
                            string[] data = DataList[index].Split(',');
                            Console.WriteLine("\n YOUR CURRENT BALANCE IS RAND : {0} ", data[2]);
                        }
                        else
```



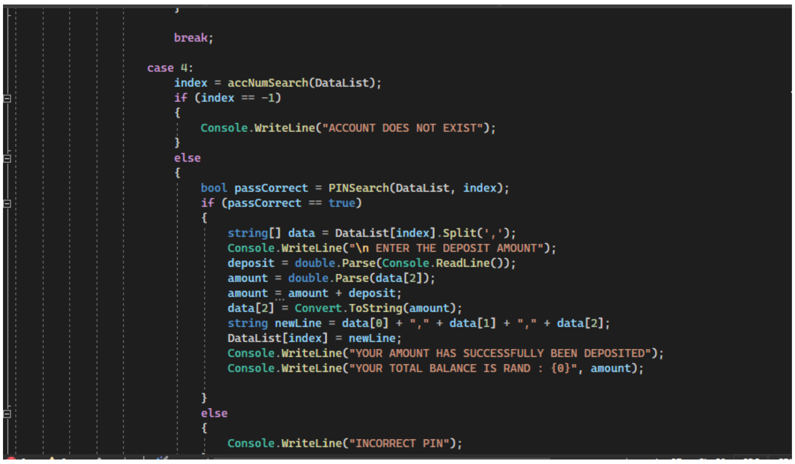
```
        else
        {
            bool passCorrect = PINSearch(DataList, index);
            if(passCorrect == true)
            {
                string[] data = DataList[index].Split(',');
                Console.WriteLine("\n YOUR CURRENT BALANCE IS RAND : {0} ", data[2]);
            }
            else
            {
                Console.WriteLine("INCORRECT PIN");
            }
        }

        break;

    case 3:
        index = accNumSearch(DataList);
        if (index == -1)
        {
            Console.WriteLine("ACCOUNT DOES NOT EXIST");
        }
        else
        {
            bool passCorrect = PINSearch(DataList, index);
            if (passCorrect == true)
            {
                string[] data = DataList[index].Split(',');
                Console.WriteLine("\n ENTER THE WITHDRAW AMOUNT, MINIMUM R100: ");
                withdraw = double.Parse(Console.ReadLine());
```



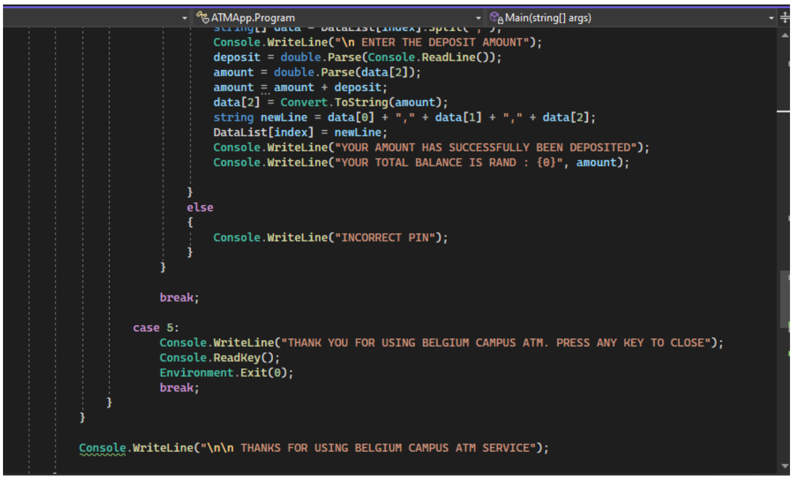
```
        }

        break;

    case 3:
        index = accNumSearch(DataList);
        if (index == -1)
        {
            Console.WriteLine("ACCOUNT DOES NOT EXIST");
        }
        else
        {
            bool passCorrect = PINSearch(DataList, index);
            if (passCorrect == true)
            {
                string[] data = DataList[index].Split(',');
                Console.WriteLine("\n ENTER THE WITHDRAW AMOUNT, MINIMUM R100: ");
                withdraw = double.Parse(Console.ReadLine());

                if (withdraw < 100 )
                {
                    Console.WriteLine("\n PLEASE ENTER A AMOUNT ABOVE R100");
                }
                else if (withdraw > (amount - 1000))
                {
                    Console.WriteLine("\n SORRY! INSUFFICENT BALANCE. MINIMUM BALANCE IS R1000");
                }
                else
                {
                    double bal = double.Parse(data[2]);
```



```
        }

        break;

    case 4:
        index = accNumSearch(DataList);
        if (index == -1)
        {
            Console.WriteLine("ACCOUNT DOES NOT EXIST");
        }
        else
        {
            bool passCorrect = PINSearch(DataList, index);
            if (passCorrect == true)
            {
                string[] data = DataList[index].Split(',');
                Console.WriteLine("\n ENTER THE DEPOSIT AMOUNT");
                deposit = double.Parse(Console.ReadLine());
                amount = double.Parse(data[2]);
                amount = amount + deposit;
                data[2] = Convert.ToString(amount);
                string newLine = data[0] + "," + data[1] + "," + data[2];
                DataList[index] = newLine;
                Console.WriteLine("YOUR AMOUNT HAS SUCCESSFULLY BEEN DEPOSITED");
                Console.WriteLine("YOUR TOTAL BALANCE IS RAND : {0}", amount);

            }
            else
            {
                Console.WriteLine("INCORRECT PIN");
```



```
            }

            break;

        case 4:
            index = accNumSearch(DataList);
            if (index == -1)
            {
                Console.WriteLine("ACCOUNT DOES NOT EXIST");
            }
            else
            {
                bool passCorrect = PINSearch(DataList, index);
                if (passCorrect == true)
                {
                    string[] data = DataList[index].Split(',');
                    Console.WriteLine("\n ENTER THE DEPOSIT AMOUNT");
                    deposit = double.Parse(Console.ReadLine());
                    amount = double.Parse(data[2]);
                    amount = amount + deposit;
                    data[2] = Convert.ToString(amount);
                    string newLine = data[0] + "," + data[1] + "," + data[2];
                    DataList[index] = newLine;
                    Console.WriteLine("YOUR AMOUNT HAS SUCCESSFULLY BEEN DEPOSITED");
                    Console.WriteLine("YOUR TOTAL BALANCE IS RAND : {0}", amount);

                }
                else
                {
                    Console.WriteLine("INCORRECT PIN");
```



```
                    string[] data = DataList[index].Split(',');
                    Console.WriteLine("\n ENTER THE DEPOSIT AMOUNT");
                    deposit = double.Parse(Console.ReadLine());
                    amount = double.Parse(data[2]);
                    amount = amount + deposit;
                    data[2] = Convert.ToString(amount);
                    string newLine = data[0] + "," + data[1] + "," + data[2];
                    DataList[index] = newLine;
                    Console.WriteLine("YOUR AMOUNT HAS SUCCESSFULLY BEEN DEPOSITED");
                    Console.WriteLine("YOUR TOTAL BALANCE IS RAND : {0}", amount);

                }
                else
                {
                    Console.WriteLine("INCORRECT PIN");
                }
            }

            break;

        case 5:
            Console.WriteLine("THANK YOU FOR USING BELGIUM CAMPUS ATM. PRESS ANY KEY TO CLOSE");
            Console.ReadKey();
            Environment.Exit(0);
            break;
    }
}

Console.WriteLine("\n\n THANKS FOR USING BELGIUM CAMPUS ATM SERVICE");
```

We have made use of a while loop, and if else statements and data types such as int to store the value set for the available balance. We have integrated the menu to work with a switch case. The menu as shown above has 5 choices. The 5th choice will close the console app and output a “Thank you message” once the account holder decides to exit the application. The \n” after each console.writeline allows a space after each option. We have integrated the console app to allow multiple accounts and we have also included lists to work with the console app. We have also created methods using Public Static Int and public static bool and integrated a for loop to search through the accountNum and return a value. The console app also makes use of a bool to verify whether the pin inserted is true or false.

Data Types and variable declarations used:

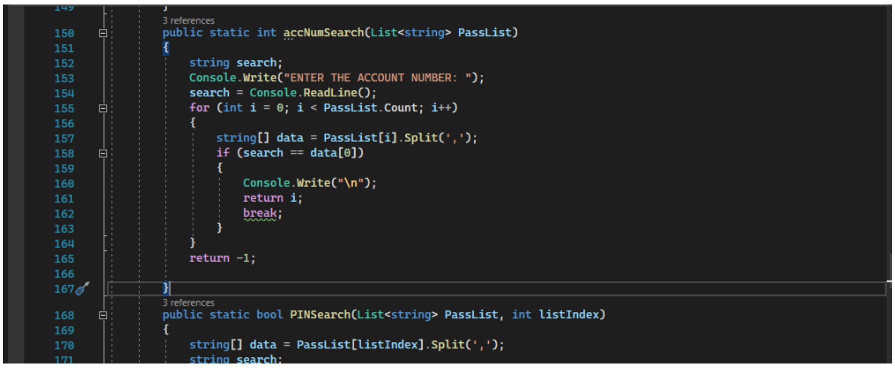
String: To store text

Int: To store numeric values

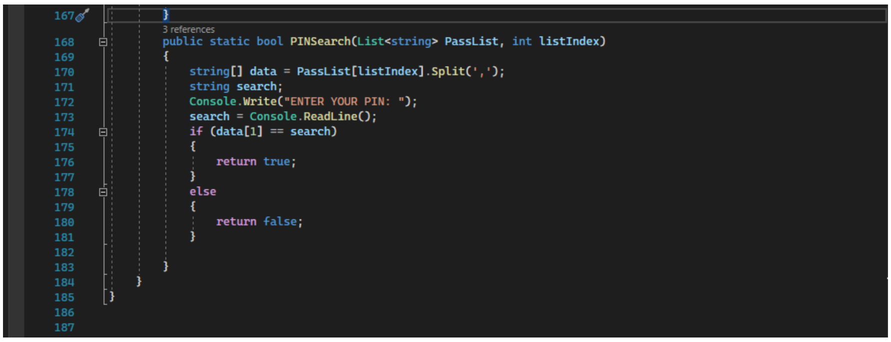
Bool: To check pin inserted is correct

Double: To store accountNum and Pin, withdraw

Methods and lists for search account number and Pin functions:



```
public static int accNumSearch(List<string> PassList)
{
    string search;
    Console.Write("ENTER THE ACCOUNT NUMBER: ");
    search = Console.ReadLine();
    for (int i = 0; i < PassList.Count; i++)
    {
        string[] data = PassList[i].Split(',');
        if (search == data[0])
        {
            Console.Write("\n");
            return i;
            break;
        }
    }
    return -1;

}
public static bool PINSearch(List<string> PassList, int listIndex)
{
    string[] data = PassList[listIndex].Split(',');
    string search;
```



```
}
public static bool PINSearch(List<string> PassList, int listIndex)
{
    string[] data = PassList[listIndex].Split(',');
    string search;
    Console.Write("ENTER YOUR PIN: ");
    search = Console.ReadLine();
    if (data[1] == search)
    {
        return true;
    }
    else
    {
        return false;
    }

}
    }
}
```

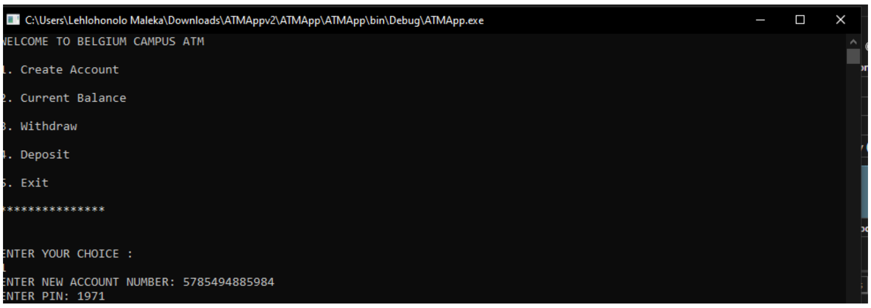
Running Console App Output:

The following output is shown once the console app has been run.

WELCOME TO BELGIUM CAMPUS ATM SERVICE

- Create Account
- Current Balance
- Withdraw
- Deposit
- Exit

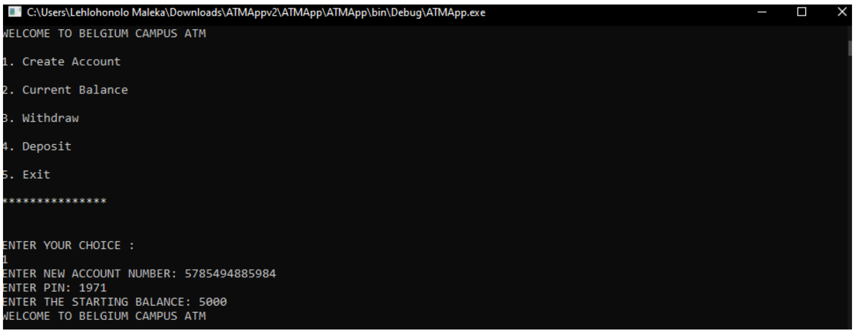
Create Account: The console app will prompt the user to create an account by entering their account number as well as their pin



C:\Users\Lehlohonolo Maleka\Downloads\ATMAppv2\ATMApp\ATMApp\bin\Debug\ATMApp.exe

```
WELCOME TO BELGIUM CAMPUS ATM

1. Create Account

2. Current Balance

3. Withdraw

4. Deposit

5. Exit

***************


ENTER YOUR CHOICE :
1
ENTER NEW ACCOUNT NUMBER: 5785494885984
ENTER PIN: 1971
```

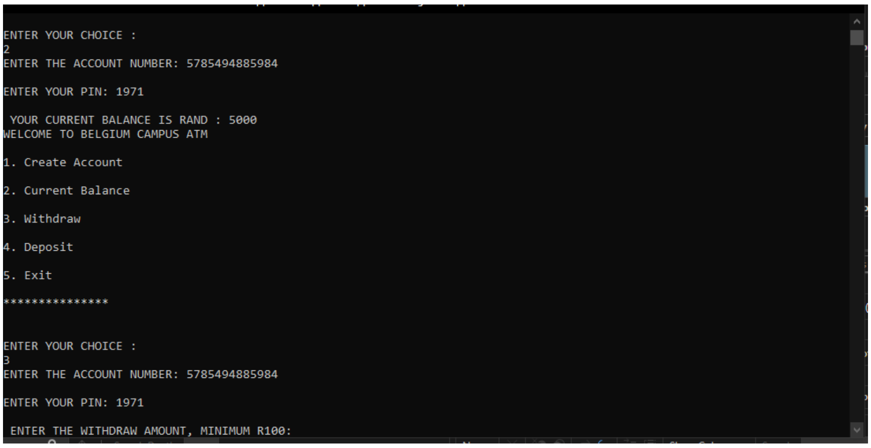
Balance: Once the account has been accessed, the Console app allows the user to set a balance The available balance has been set to R5000



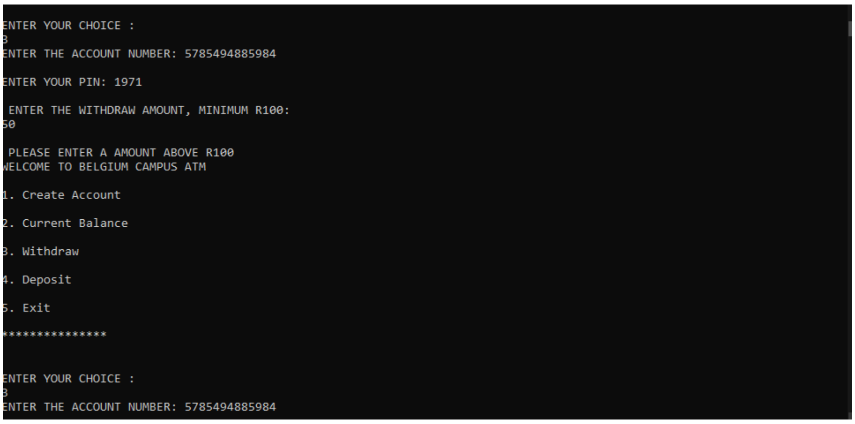
C:\Users\Lehlohonolo Maleka\Downloads\ATMAppv2\ATMApp\ATMApp\bin\Debug\ATMApp.exe

```
WELCOME TO BELGIUM CAMPUS ATM

1. Create Account

2. Current Balance

3. Withdraw

4. Deposit

5. Exit

**************


ENTER YOUR CHOICE :
1
ENTER NEW ACCOUNT NUMBER: 5785494885984
ENTER PIN: 1971
ENTER THE STARTING BALANCE: 5000
WELCOME TO BELGIUM CAMPUS ATM
```

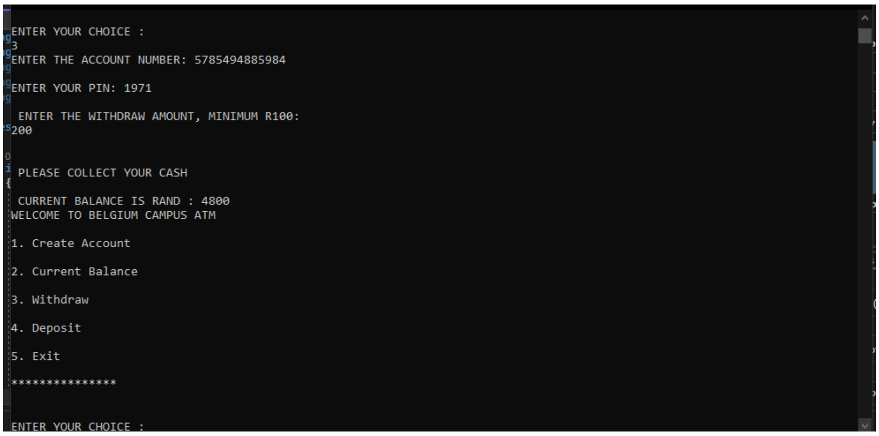
Withdrawal: Depending on the withdrawal amount the app will return the balance left after a user withdraws, since the minimal withdrawal amount is above R100 if a user withdraws anything less than R100 the console app will display “enter an amount above R100, and the remaining balance will be displayed as shown in the below images



```
ENTER YOUR CHOICE :
2
ENTER THE ACCOUNT NUMBER: 5785494885984

ENTER YOUR PIN: 1971

 YOUR CURRENT BALANCE IS RAND : 5000
WELCOME TO BELGIUM CAMPUS ATM

1. Create Account

2. Current Balance

3. Withdraw

4. Deposit

5. Exit

**************


ENTER YOUR CHOICE :
3
ENTER THE ACCOUNT NUMBER: 5785494885984

ENTER YOUR PIN: 1971

 ENTER THE WITHDRAW AMOUNT, MINIMUM R100:
```

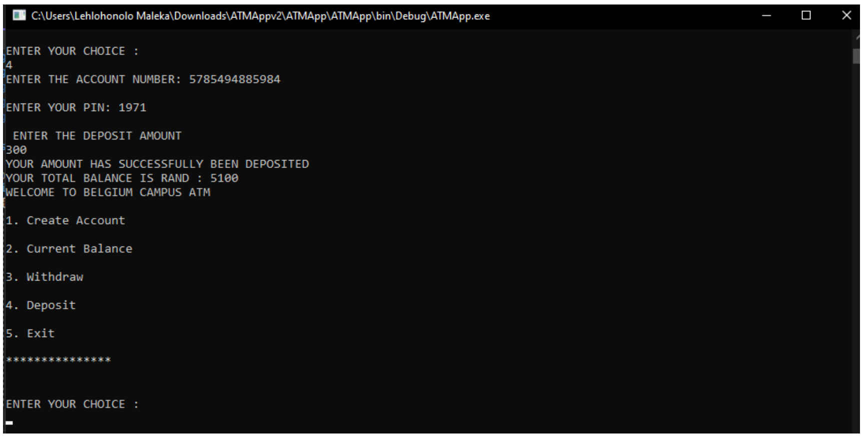


```
ENTER YOUR CHOICE :
3
ENTER THE ACCOUNT NUMBER: 5785494885984

ENTER YOUR PIN: 1971

 ENTER THE WITHDRAW AMOUNT, MINIMUM R100:
50

 PLEASE ENTER A AMOUNT ABOVE R100
WELCOME TO BELGIUM CAMPUS ATM

1. Create Account

2. Current Balance

3. Withdraw

4. Deposit

5. Exit

**************


ENTER YOUR CHOICE :
3
ENTER THE ACCOUNT NUMBER: 5785494885984
```



```
ENTER YOUR CHOICE :
3
ENTER THE ACCOUNT NUMBER: 5785494885984

ENTER YOUR PIN: 1971

 ENTER THE WITHDRAW AMOUNT, MINIMUM R100:
200


 PLEASE COLLECT YOUR CASH

 CURRENT BALANCE IS RAND : 4800
WELCOME TO BELGIUM CAMPUS ATM

1. Create Account

2. Current Balance

3. Withdraw

4. Deposit

5. Exit

***************


ENTER YOUR CHOICE :
```

Deposit: The console app allows the user to deposit an amount. If the account holder deposits an amount that is more than what is in the current balance, their balance will increase. Once the money has been deposited the console app will display “You have deposited your money successfully”.



C:\Users\Lehlohonolo Maleka\Downloads\ATMAppv2\ATMApp\ATMApp\bin\Debug\ATMApp.exe

```
ENTER YOUR CHOICE :
4
ENTER THE ACCOUNT NUMBER: 5785494885984

ENTER YOUR PIN: 1971

 ENTER THE DEPOSIT AMOUNT
300
YOUR AMOUNT HAS SUCCESSFULLY BEEN DEPOSITED
YOUR TOTAL BALANCE IS RAND : 5100
WELCOME TO BELGIUM CAMPUS ATM

1. Create Account

2. Current Balance

3. Withdraw

4. Deposit

5. Exit

***************


ENTER YOUR CHOICE :
```