```java
package com.atm;
import java.lang.Math;
import java.util.Random;

public class Atm {

	//instance variables

	private int accountNumber;
	private String customerName;
	private String accountType;
	private double balance;
	private double minimumBalance;
	private long mobNumber;
	private String emailId;
	private int atmPin;

	//default constructor
	public Atm() {
		super();
		// TODO Auto-generated constructor stub
	}

	//parameterized constructor
	public Atm(int accountNumber, String customerName, String accountType, double 
balance, double minimumBalance,
			int mobNumber, String emailId, int atmPin) {
		super();

		this.accountNumber = accountNumber;
		this.customerName = customerName;
		this.accountType = accountType;
		this.balance = balance;
		this.minimumBalance = minimumBalance;
		this.mobNumber = mobNumber;
		this.emailId = emailId;
		this.atmPin = atmPin;
	}


	public int getAccountNumber() {
		return accountNumber;
	}

	public void setAccountNumber(int accountNumber) {
		this.accountNumber = accountNumber;
	}

	public String getCustomerName() {
		return customerName;
	}
```

```java
public void setCustomerName(String customerName) {
        this.customerName = customerName;
}

public String getAccountType() {
        return accountType;
}

public void setAccountType(String accountType) {
        this.accountType = accountType;
}

public double getBalance() {
        return balance;//complete balance
        }

public void setBalance(double balance) {
        this.balance = balance;
}

public double getMinimumBalance() {
        return minimumBalance;
}

public void setMinimumBalance(double minimumBalance) {
        minimumBalance = 1000;
        this.minimumBalance = minimumBalance;
}

public long getMobNumber() {
        return mobNumber;
}

public void setMobNumber(long mobNumber) {
        this.mobNumber = mobNumber;
}

public String getEmailId() {
        return emailId;
}

public void setEmailId(String emailId) {
        this.emailId = emailId;
}

public int getAtmPin() {
        return atmPin;
}

public void setAtmPin(int atmPin) {
        this.atmPin = atmPin;
}


@Override
```

```
       public String toString() {
             return "Atm [accountNumber=" + accountNumber + ", customerName="
                          + customerName + ", accountType=" + accountType + ",
balance=" + balance + ", minimumBalance="
                          + minimumBalance + ", mobNumber=" + mobNumber + ",
emailId=" + emailId + ", atmPin=" + atmPin + "]";
       }

       public static int generateAccountNumber() {
             Random randomCustomerNumber = new Random(System.currentTimeMillis());
             return(100000000+randomCustomerNumber.nextInt(100000000));
       }

       public static int generateAtmPin() {
             Random randomAtmPin = new Random(System.currentTimeMillis());
             return(1000+randomAtmPin.nextInt(10000));
       }



}




package com.atmapp;

import java.util.ArrayList;
import java.util.Scanner;

import com.atm.Atm;
import com.atmlib.AtmLib;

public class AtmApp {
       public static void main(String[] args) {

             mainMenu();
       }

       //MainMenu for selecting the operation to be performed
       private static void mainMenu() {

             //Initializing Scanner
             Scanner input = new Scanner(System.in);

             //Initializing ArrayList

             char choiceYN= 'n';
             do {
                    //welcome page
                    System.out.println("------------Welecome--------------");
                    System.out.println("Enter the action to be performed");
                    System.out.println("1.Add Customer \n2.Update Customer details
\n3.Delete Customer \n4.Deposit Money"
```

```
                    + "\n5.Withdraw money \n6.Show Balance \n7.List of
all Customers \n8.Specific Customer details"
                    + "\n9.Tranfer money to an Account");

            int choice = input.nextInt() ;//taking the choice input

            //selecting the operation
            switch(choice) {
            case 1:
                AtmLib.addCustomer();
                break;

            case 2:
                AtmLib.updateCustomer();
                break;

            case 3:
                AtmLib.deleteCustomer();
                break;

            case 4:
                AtmLib.depositMoney();
                break;

            case 5:
                AtmLib.withdrawMoney();
                break;

            case 6:
                AtmLib.showbalance();
                break;

            case 7:
                AtmLib.customerList();
                break;

            case 8:
                AtmLib.aCustomer();
                break;

            case 9:
                AtmLib.transferMoney();
                break;

            default:
                System.out.println("Invalid entry!");
                break;

            }
            //To continue the operation
            System.out.println("Do you want to continue to Main menu (y/n)");
            choiceYN = input.next().charAt(0);
        }while(choiceYN == 'y'||choiceYN == 'Y');

    }
```

```
}


package com.atmlib;

import java.util.ArrayList;
import java.util.Scanner;

import com.atm.Atm;

public class AtmLib {

      static ArrayList<Atm> customerArray = new ArrayList<Atm>();
      //To add a customer
      public static void addCustomer(){

             char choice = 'n';


         do {
              Scanner objScanner = new Scanner(System.in);
              Atm customer = new Atm();


             System.out.println("Enter the customer name: ");
             customer.setCustomerName(objScanner.nextLine());


             System.out.println("Enter the customer account type: \n1. Savings\n2.
Current");
              switch (Integer.parseInt(objScanner.next())) {
              case 1:
                  customer.setAccountType("Savings");
                  break;
              case 2:
                  customer.setAccountType("Fixed");
                  break;
              default:
                  System.out.println("Invalid entry! please try again");
              }


             //customer mobile number
             System.out.println("Enter the customer mobile number: ");
             customer.setMobNumber(objScanner.nextLong());

             //taking email
             System.out.println("Enter the customer email id: ");
             customer.setEmailId(objScanner.next());

             //Atm pin generator
             customer.setAccountNumber(Atm.generateAccountNumber());
             customer.setAtmPin(Atm.generateAtmPin());
```

```
              //Taking the balance input
              System.out.println("Account is created with a minimum balance of
1000.00 ");
              customer.setBalance(1000);

              customerArray.add(customer);
               System.out.println("Your account has been added successfully\n");
              System.out.println(customer);


               System.out.println("Do you want to add another record..? y or n");
               choice = objScanner.next().charAt(0);
           } while (choice == 'y' || choice == 'Y');
       }


       //To Update customer details
       public static void updateCustomer(){

             //Initializing scanner
             Scanner input = new Scanner(System.in);

             System.out.println("Enter the 9 digit account number");
             long accountNo = input.nextLong();

             for (Atm customer : customerArray) {
                    if (customer.getAccountNumber()==accountNo) {

            Scanner scanner = new Scanner(System.in);
               int option ;
               System.out.println("Enter the field to update: \n1.Mobile
Number\n2.Email Id");
               switch(Integer.parseInt(scanner.next())){
               case 1:
                    System.out.println("Enter the mobile number: ");
                   customer.setMobNumber(Integer.parseInt(scanner.next()));
                   break;
               case 2:
                    System.out.println("Enter the email id: ");
                   customer.setEmailId(scanner.next());
                   break;

               default:
                   System.out.println("Invalid input!!");
                   break;
                    }
         }
      }
       }


       //To delete a customer
       public static void deleteCustomer(){

             //Initializing scanner
```

```
                          Scanner input = new Scanner(System.in);

                          //Taking the account number as input
                          System.out.println("Enter the 9 digit account number");
                          long accountNo = input.nextLong();

                          //scanning for the given account number in the
customrArray
                          for (Atm customer : customerArray) {
                                 if (customer.getAccountNumber()==accountNo) {

                                        customer.setAccountNumber(0);
                                        customer.setAtmPin(0);
                                customer.setAccountType(null);
                                customer.setCustomerName(null);
                                customer.setBalance(input.nextDouble());
                                customer.setEmailId(null);
                                customer.setMinimumBalance(0.00);
                                customer.setMobNumber(0000000000);
                                System.out.println("The customer details are
deleted");

                                 }
                                 else if(customer.getAccountNumber()!=accountNo) {
                                        System.out.println("This bank account doesn't
exit");
                                 }
                    }
       }


       //To deposit money
       public static void depositMoney(){

             //Initializing scanner
             Scanner input = new Scanner(System.in);

             //Taking the account number as input
             System.out.println("Enter the 9 digit account number");
             long accountNo = input.nextLong();

             //scanning for the given account number in the customrArray
             for (Atm customer : customerArray) {
                    if (customer.getAccountNumber()==accountNo) {

                          System.out.println("Enter the amount to be deposited");
                          double amountDeposited = input.nextDouble();

                          if(amountDeposited > 50000) {
                                 System.out.println("Enter your pan card number");
                                 long panNumber = input.nextLong();

       customer.setBalance(customer.getBalance()+amountDeposited);
                                 System.out.println("The new Balance of the account
number"+ accountNo+" is "+customer.getBalance());
```

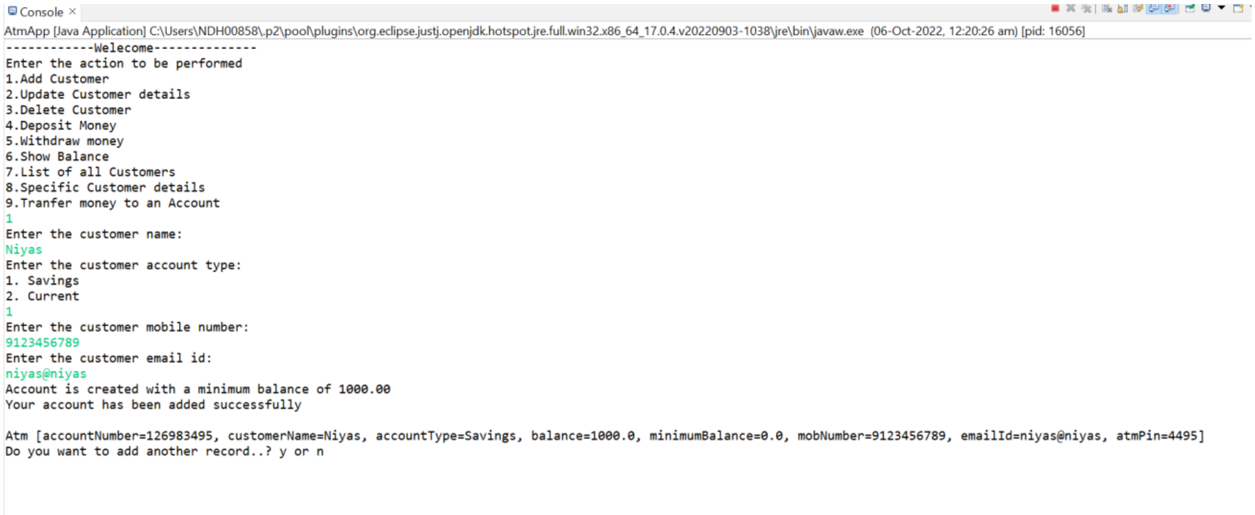
```
                }

                else if (amountDeposited >0 && amountDeposited<50000 ) {

    customer.setBalance(customer.getBalance()+amountDeposited);
                }
            }
    }
    }


    //To withdraw money
    public static void withdrawMoney(){

        //Initializing scanner
        Scanner input = new Scanner(System.in);

        //Taking the account number as input
        System.out.println("Enter the 9 digit account number");
        long accountNo = input.nextLong();

        //scanning for the given account number in the customrArray
        for (Atm customer : customerArray) {
            if (customer.getAccountNumber()==accountNo) {

                System.out.println("Enter the amount to withdraw");
                double amountToWithdraw = input.nextDouble();

                if(amountToWithdraw > 50000) {
                    System.out.println("Enter your pan card number");
                    long panNumber = input.nextLong();
                    customer.setBalance(customer.getBalance()-
amountToWithdraw);
                    System.out.println("The new Balance of the account
number"+ accountNo+" is "+customer.getBalance());
                }

                else if (amountToWithdraw >0 && amountToWithdraw<50000 ) {
                    customer.setBalance(customer.getBalance()-
amountToWithdraw);
                    System.out.println("The new Balance of the account
number"+ accountNo+" is "+customer.getBalance());
                }
            }
        }
    }


    //To show balance of a customer
    public static void showbalance() {

        //Initializing scanner
        Scanner input = new Scanner(System.in);

        //Taking the account number as input
```

```
             System.out.println("Enter the 9 digit account number");
             long accountNo = input.nextLong();

             //scanning for the given account number in the customrArray
             for (Atm customer : customerArray) {
                   if (customer.getAccountNumber()==accountNo) {

                          System.out.println("The balace amount that is available to
withdraw is  "+customer.getBalance());

         }
      }
       }


       //To display customer list
       public static void customerList() {

             for (Atm customer : customerArray) {
                   System.out.println(customerArray);
             }
       }


       //To show details of a specific customer
       public static void aCustomer() {

             //Initializing scanner
             Scanner input = new Scanner(System.in);

             //Taking the account number as input
             System.out.println("Enter the 9 digit account number");
             long accountNo = input.nextLong();

             //scanning for the given account number in the customrArray
             for (Atm customer : customerArray) {
                   if (customer.getAccountNumber()==accountNo) {

                          System.out.println(customer.getCustomerName());
                          System.out.println("Account number
"+customer.getAccountNumber());
                          System.out.println("Account type
"+customer.getAccountType());
                          System.out.println("Account Pin "+customer.getAtmPin());
                          System.out.println("The Balance amount excluding minimum
balance is  "+ customer.getBalance());
                          System.out.println("Minimum balance
"+customer.getMinimumBalance());
                          System.out.println("Email Id  "+customer.getEmailId());
                          System.out.println("Mob number "+customer.getMobNumber());
         }

      }

       }
```

```
    //Transfer money from one account to another
    public static void transferMoney() {

        //Initializing scanner
        Scanner input = new Scanner(System.in);

        //Taking the account number as input
        System.out.println("Enter the 9 digit account number ");
        long accountNoToWithdraw = input.nextLong();

        //Taking the account number of the receiving Bankaccount
        System.out.println("Enter the account number of the  receiving Bank
account");
        long accountNoToDeposit = input.nextLong();

        //scanning for the given account number in the customrArray
        for (Atm customer : customerArray) {
            if(customer.getAccountNumber()==accountNoToDeposit) {

                if (customer.getAccountNumber()==accountNoToWithdraw) {

                    System.out.println("Enter the amount to transfer");
                    double amountToTransfer = input.nextDouble();

                    if(amountToTransfer<customer.getBalance()) {
                        System.out.println("Insufficeint funds!");
                    }

                    else {
                        if(amountToTransfer > 50000) {
                            System.out.println("Enter your pan card
number");
                            long panNumber = input.nextLong();

    customer.setBalance(customer.getBalance()-amountToTransfer);
                            System.out.println("The new Balance of
the account number"+ accountNoToWithdraw+" is "+customer.getBalance());

                            //Setting the balance of the receiving
Bank account

    if(customer.getAccountNumber()==accountNoToDeposit) {

    customer.setBalance(customer.getBalance()+amountToTransfer);
                                System.out.println("The new
Balance of the account number"+ accountNoToDeposit+" is "+customer.getBalance());
                            }
                        }

                        else if (amountToTransfer >0 &&
amountToTransfer<50000 ) {

    customer.setBalance(customer.getBalance()-amountToTransfer);
```

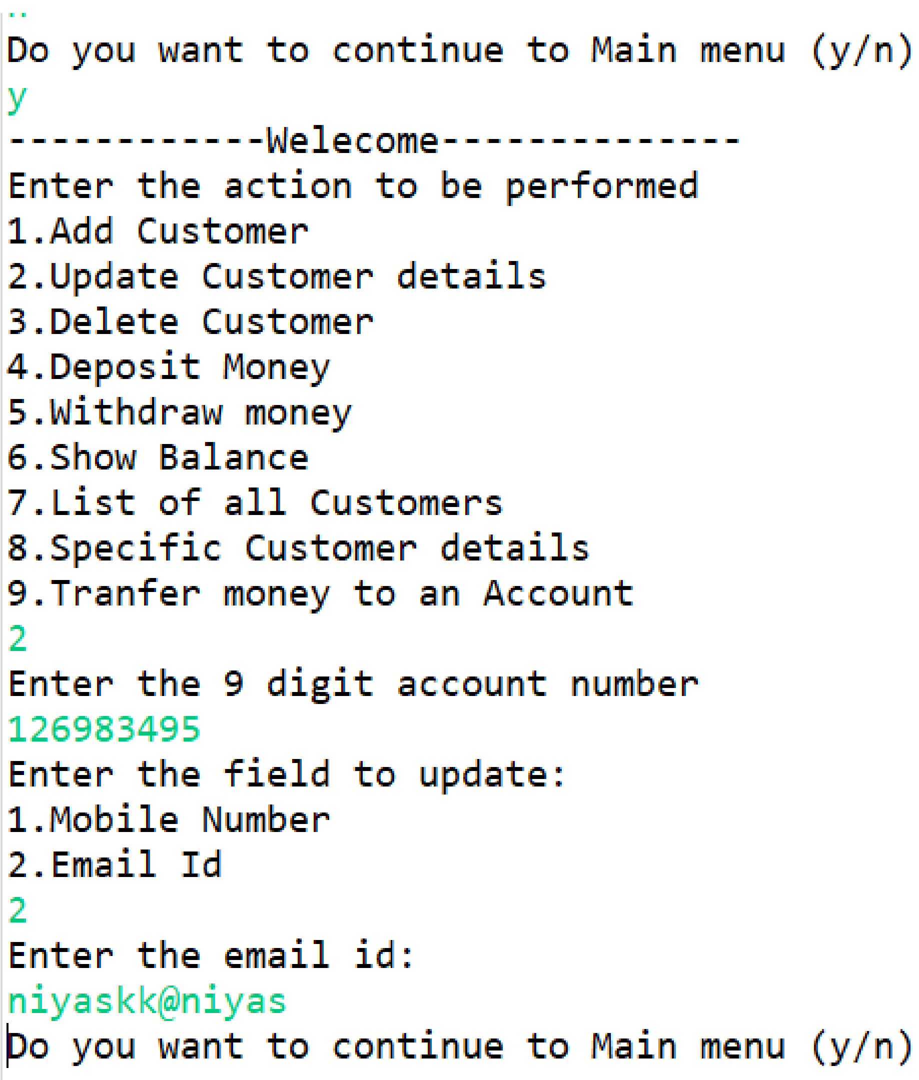
```
                                        }
                                 }
                    }
                          else {
                                 System.out.println("There is no receiving bank
account in this account number");
                          }


            }

         }

       }

}
```

Output 1



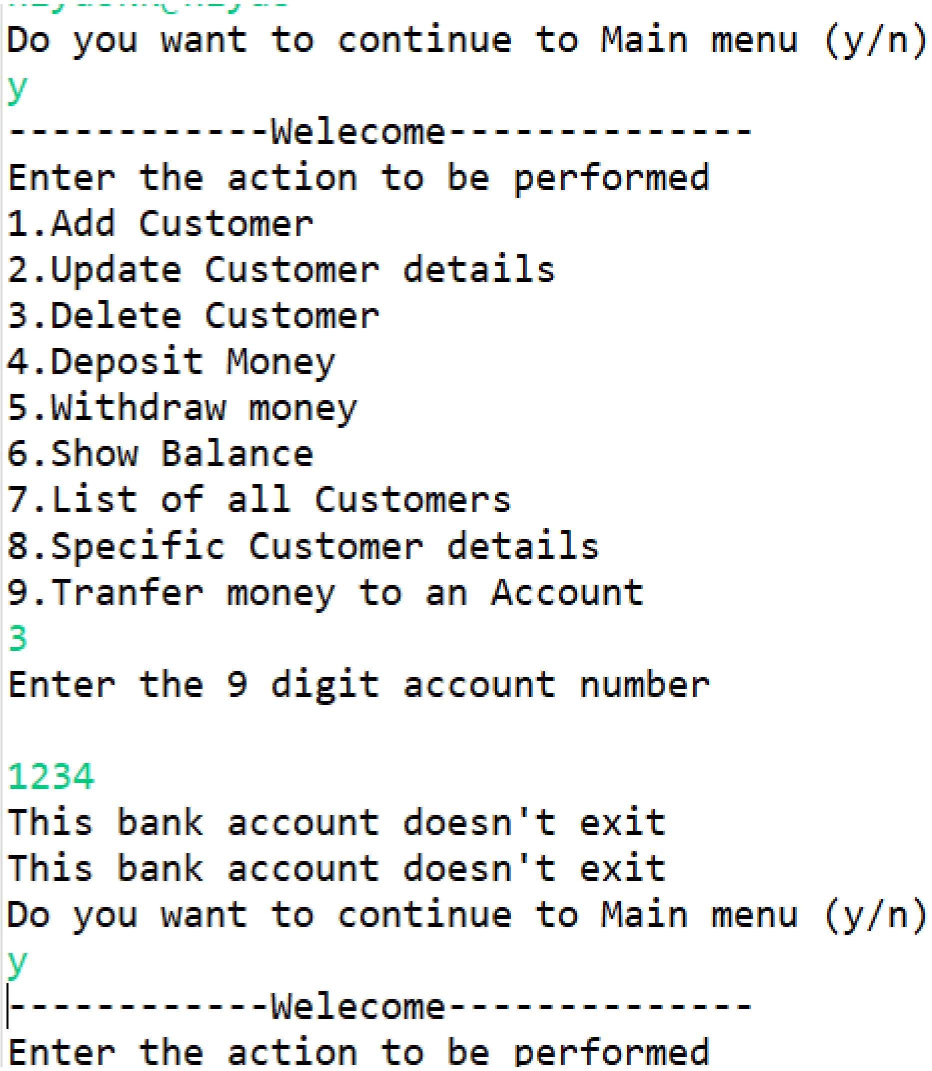
```
AtmApp [Java Application] C:\Users\NDH00858\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220903-1038\jre\bin\javaw.exe  (06-Oct-2022, 12:20:26 am) [pid: 16056]
------------Welecome--------------
Enter the action to be performed
1.Add Customer
2.Update Customer details
3.Delete Customer
4.Deposit Money
5.Withdraw money
6.Show Balance
7.List of all Customers
8.Specific Customer details
9.Tranfer money to an Account
1
Enter the customer name:
Niyas
Enter the customer account type:
1. Savings
2. Current
1
Enter the customer mobile number:
9123456789
Enter the customer email id:
niyas@niyas
Account is created with a minimum balance of 1000.00
Your account has been added successfully

Atm [accountNumber=126983495, customerName=Niyas, accountType=Savings, balance=1000.0, minimumBalance=0.0, mobNumber=9123456789, emailId=niyas@niyas, atmPin=4495]
Do you want to add another record..? y or n
```

Output 2



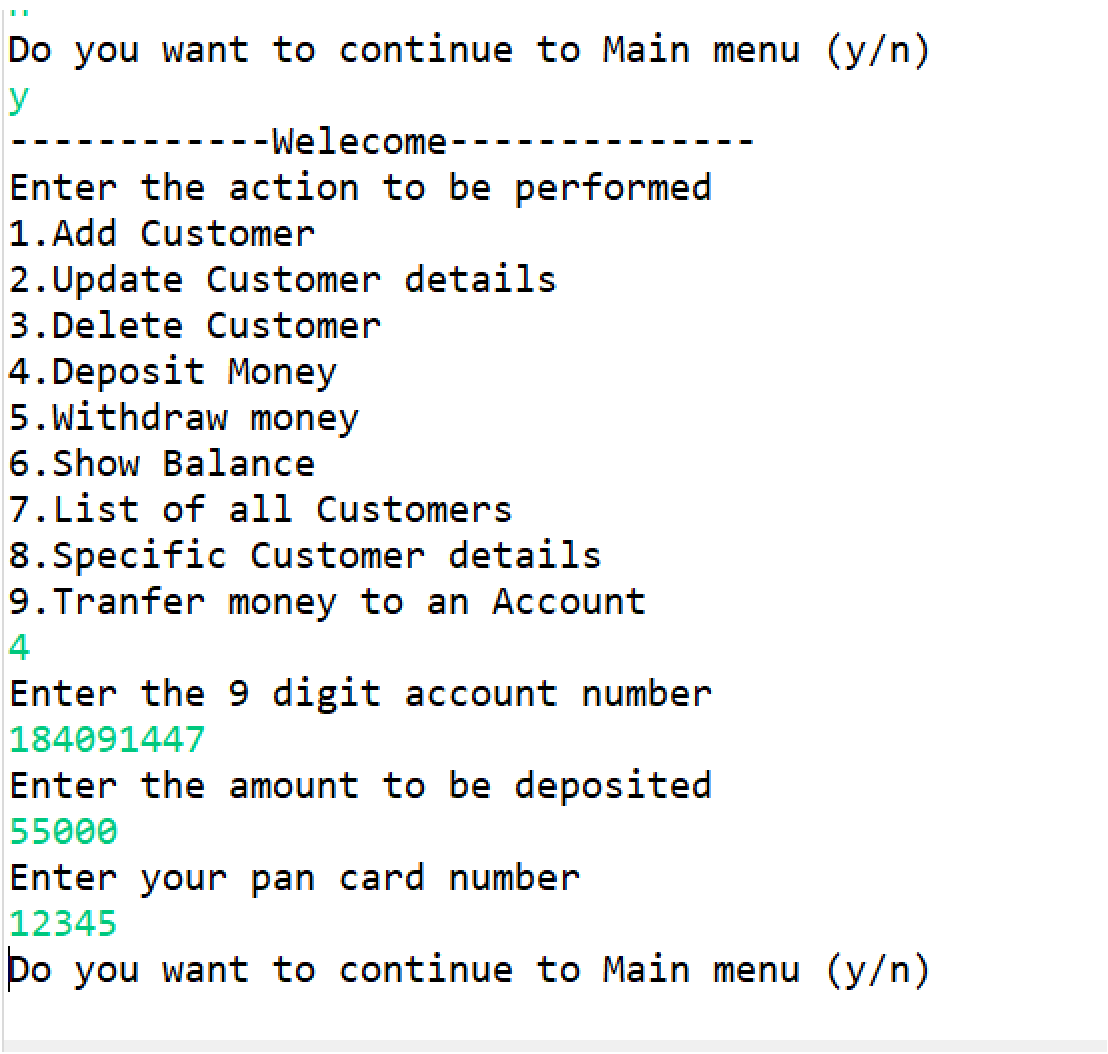
```
Do you want to continue to Main menu (y/n)
y
------------Welecome--------------
Enter the action to be performed
1.Add Customer
2.Update Customer details
3.Delete Customer
4.Deposit Money
5.Withdraw money
6.Show Balance
7.List of all Customers
8.Specific Customer details
9.Tranfer money to an Account
2
Enter the 9 digit account number
126983495
Enter the field to update:
1.Mobile Number
2.Email Id
2
Enter the email id:
niyaskk@niyas
Do you want to continue to Main menu (y/n)
```

Output 3



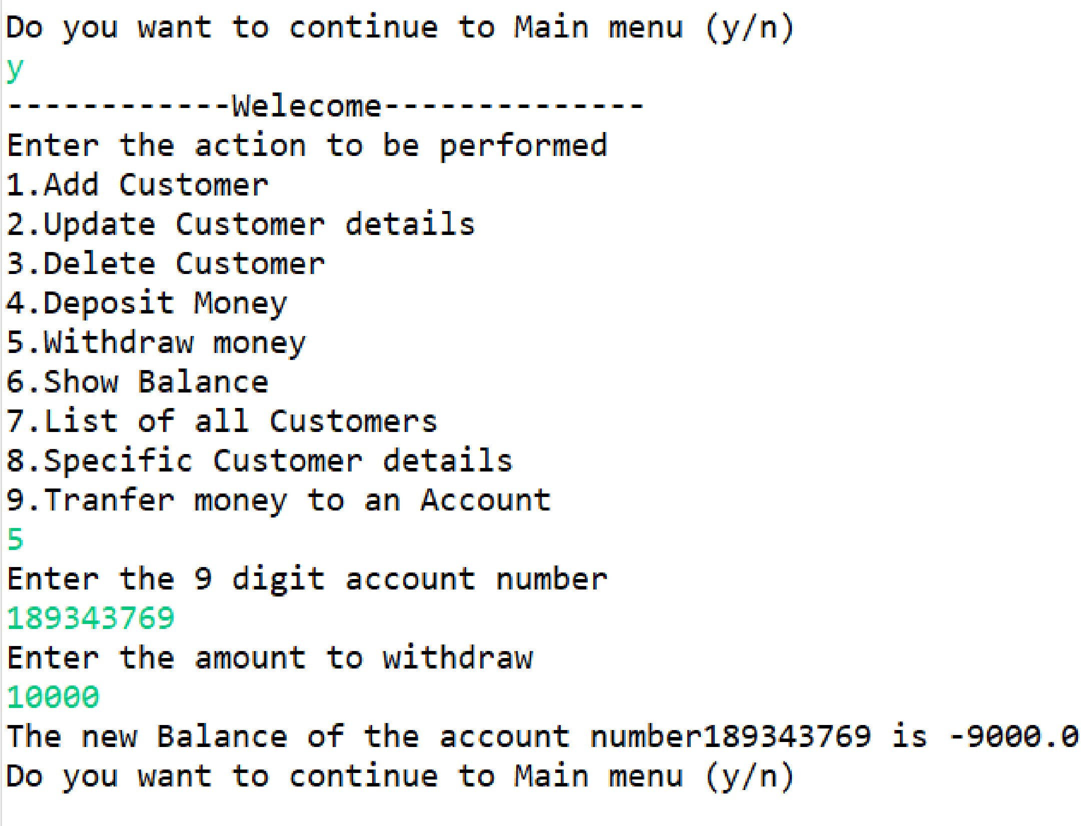
```
Do you want to continue to Main menu (y/n)
y
------------Welecome--------------
Enter the action to be performed
1.Add Customer
2.Update Customer details
3.Delete Customer
4.Deposit Money
5.Withdraw money
6.Show Balance
7.List of all Customers
8.Specific Customer details
9.Tranfer money to an Account
3
Enter the 9 digit account number

1234
This bank account doesn't exit
This bank account doesn't exit
Do you want to continue to Main menu (y/n)
y
------------Welecome--------------
Enter the action to be performed
```

Output 4



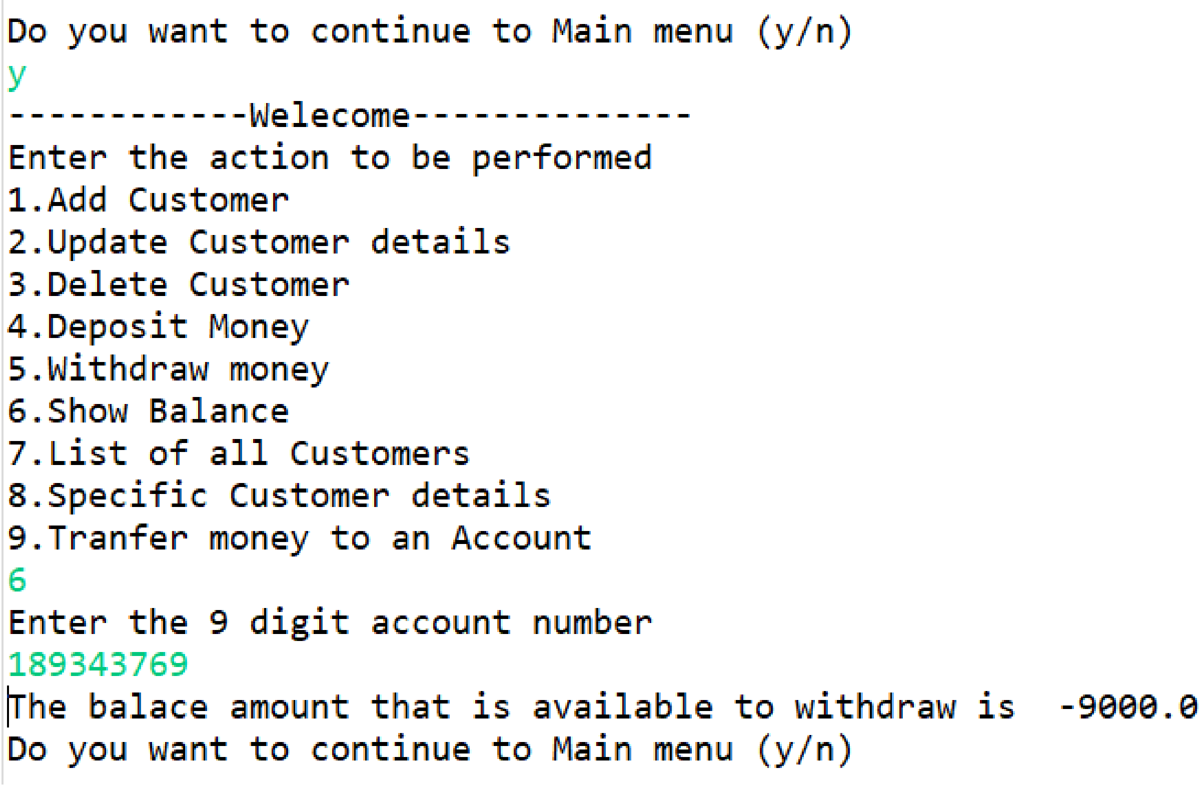
```
Do you want to continue to Main menu (y/n)
y
------------Welecome--------------
Enter the action to be performed
1.Add Customer
2.Update Customer details
3.Delete Customer
4.Deposit Money
5.Withdraw money
6.Show Balance
7.List of all Customers
8.Specific Customer details
9.Tranfer money to an Account
4
Enter the 9 digit account number
184091447
Enter the amount to be deposited
55000
Enter your pan card number
12345
Do you want to continue to Main menu (y/n)
```

Output 5



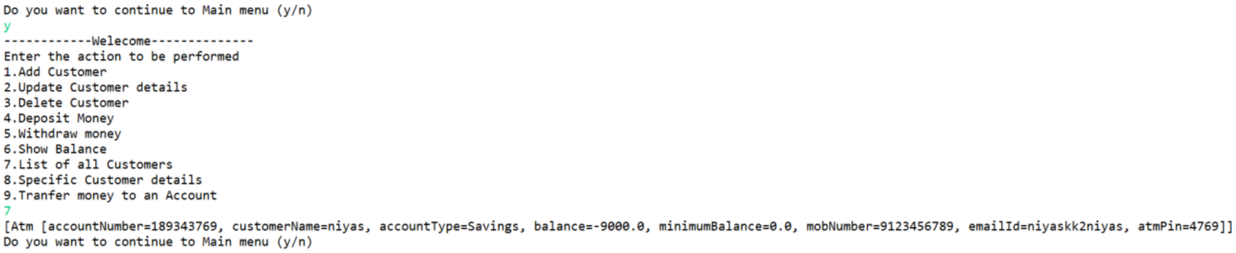
```
Do you want to continue to Main menu (y/n)
y
------------Welecome--------------
Enter the action to be performed
1.Add Customer
2.Update Customer details
3.Delete Customer
4.Deposit Money
5.Withdraw money
6.Show Balance
7.List of all Customers
8.Specific Customer details
9.Tranfer money to an Account
5
Enter the 9 digit account number
189343769
Enter the amount to withdraw
10000
The new Balance of the account number189343769 is -9000.0
Do you want to continue to Main menu (y/n)
```

Output 6



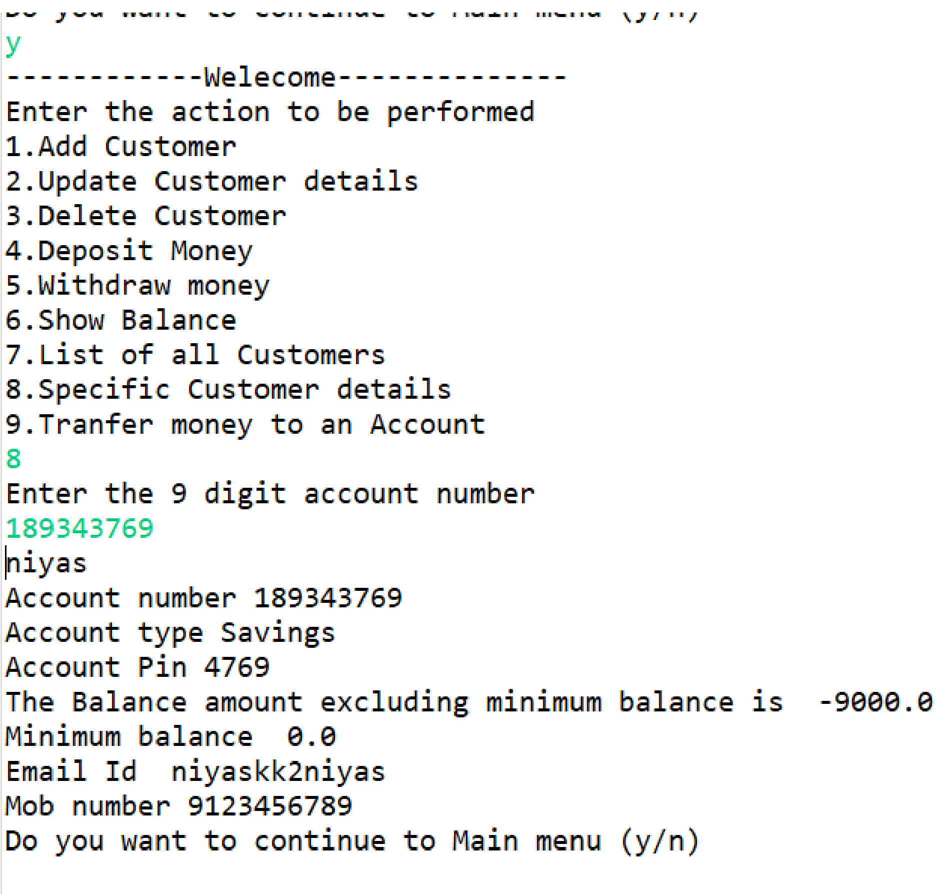
```
Do you want to continue to Main menu (y/n)
y
------------Welecome--------------
Enter the action to be performed
1.Add Customer
2.Update Customer details
3.Delete Customer
4.Deposit Money
5.Withdraw money
6.Show Balance
7.List of all Customers
8.Specific Customer details
9.Tranfer money to an Account
6
Enter the 9 digit account number
189343769
The balace amount that is available to withdraw is  -9000.0
Do you want to continue to Main menu (y/n)
```

Output 7



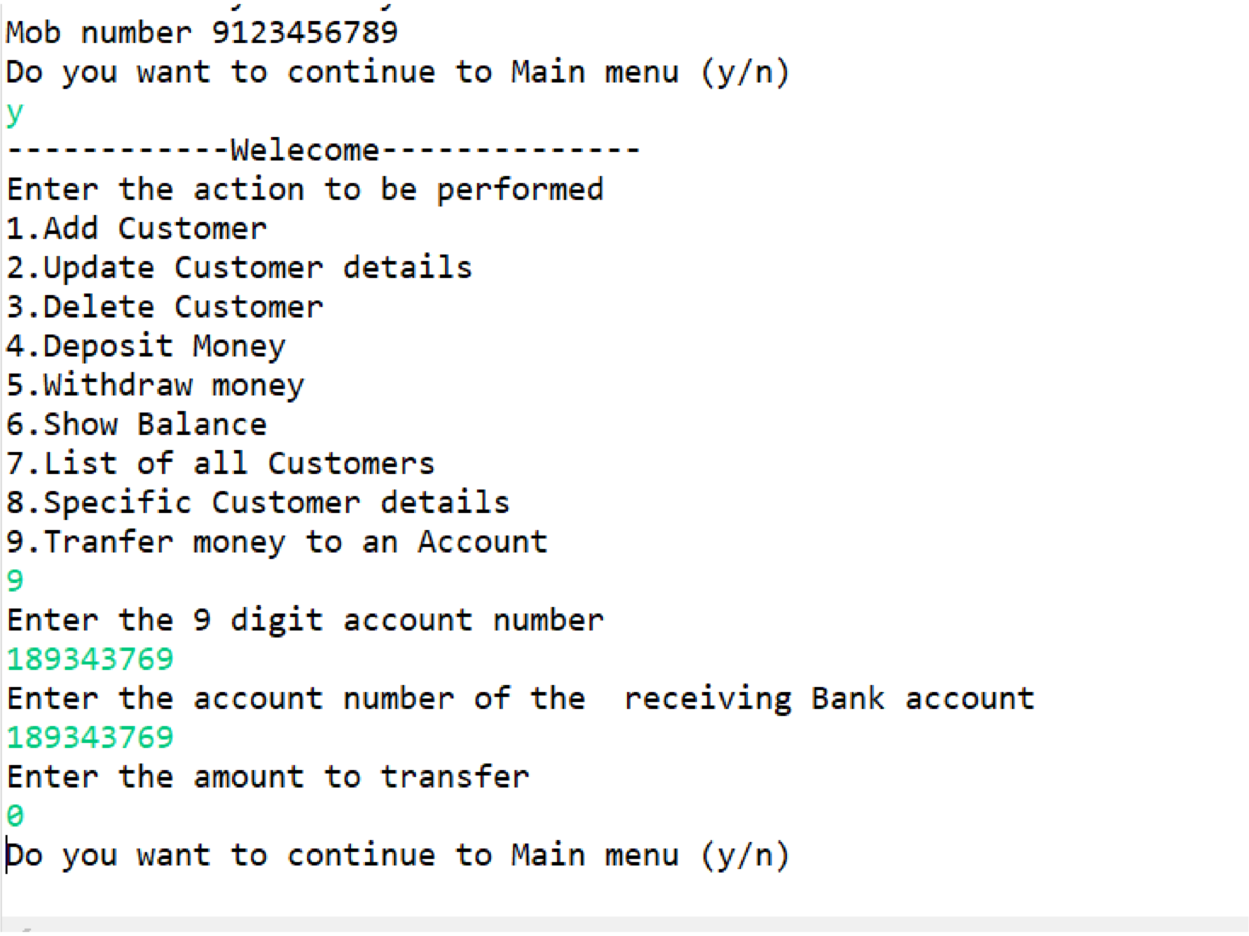
```
Do you want to continue to Main menu (y/n)
y
------------Welecome--------------
Enter the action to be performed
1.Add Customer
2.Update Customer details
3.Delete Customer
4.Deposit Money
5.Withdraw money
6.Show Balance
7.List of all Customers
8.Specific Customer details
9.Tranfer money to an Account
7
[Atm [accountNumber=189343769, customerName=niyas, accountType=Savings, balance=-9000.0, minimumBalance=0.0, mobNumber=9123456789, emailId=niyaskk2niyas, atmPin=4769]]
Do you want to continue to Main menu (y/n)
```

Output 8



```
Do you want to continue to Main menu (y/n)
y
------------Welecome--------------
Enter the action to be performed
1.Add Customer
2.Update Customer details
3.Delete Customer
4.Deposit Money
5.Withdraw money
6.Show Balance
7.List of all Customers
8.Specific Customer details
9.Tranfer money to an Account
8
Enter the 9 digit account number
189343769
niyas
Account number 189343769
Account type Savings
Account Pin 4769
The Balance amount excluding minimum balance is  -9000.0
Minimum balance  0.0
Email Id  niyaskk2niyas
Mob number 9123456789
Do you want to continue to Main menu (y/n)
```

Output 9



```
Mob number 9123456789
Do you want to continue to Main menu (y/n)
y
------------Welecome--------------
Enter the action to be performed
1.Add Customer
2.Update Customer details
3.Delete Customer
4.Deposit Money
5.Withdraw money
6.Show Balance
7.List of all Customers
8.Specific Customer details
9.Tranfer money to an Account
9
Enter the 9 digit account number
189343769
Enter the account number of the  receiving Bank account
189343769
Enter the amount to transfer
0
Do you want to continue to Main menu (y/n)
```