End Term Project Report

On

Introduction to Databases (CSE 3151)

Submitted by

Name	**: Sidhanta Barik**
Reg. No.	**: 2241002049**
Branch	**: CSE, B.Tech**
Semester	**: 6th**
Section	**: 2241006**
Session	**: 2024-2025**
Admission Batch	**: 2022**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

FACULTY OF ENGINEERING & TECHNOLOGY (ITER)

SIKSHA 'O' ANUSANDHAN DEEMED TO BE UNIVERSITY

BHUBANESWAR, ODISHA – 751030

Laboratory Assignments
Subject: Introduction to Databases
Subject code: CSE 3151

Assignment 6: Mini project using JDBC connectivity

Objective of this Assignment: To design a miniature Project for a Banking Management System using Java, Oracle, and JDBC.

Requisite:

- Completion of IDB Laboratory Assignment-4
- Basic Java Programming knowledge

Overview of the Project: A Banking Management System is to be designed, putting together the concepts learnt in theory and practiced in laboratory. The Project will integrate a Java frontend menu driven program to the backend Banking Database designed in Oracle through JDBC connectivity.

Project Description: The Java program provides an interface to the user to access, insert, delete, and update the database. The program handles user input, output to and from the database for the said operations. User should be able to do the following operations:

1. Show Customer Records:
   Using this option the details of all the customers should be displayed in a particular format.

2. Add Customer Record:
   Using this option the user needs to provide the information such as cust_no, name, phoneno and city through user input, which will be saved in database. After that using option 1, details of all the customers will be displayed in a particular format.

3. Delete Customer Record:Using this option the user needs to provide the cust_no of a customer through user input and all the information related to that customer will be deleted from the database. After that using option 1, details of all the customers will be displayed in a particular format.

4. Update Customer Information:
   Using this option the user needs to provide the cust_no of a customer through user input and based on the following choice the information related to the customer will be updated.
   4.1: Update name
   4.2: Update Phoneno.
   4.3: Update city
   After that using option 1, details of all the customers will be displayed in particular format.

5. Show Account Details of a Customer:
   Using this option the user needs to provide the cust_no of a customer through user input and all the information of that customer along with his account_no, type, balance, branch_code, branch_name and branch_city will be displayed in proper format.

6. Show Loan Details of a Customer:
   Using this option the user needs to provide the cust_no of a customer through user input and all the information of that customer along with his loan_no, loan amount, branch_code, branch_name and branch_city will be displayed in proper format.

7. Deposit Money to an Account:
   Using this option the user needs to provide the account_no of a customer and the amount to be deposited through user input. According to the deposited amount the updated balance will be verified in proper format using option 5.

8. Withdraw Money from an Account:
   Using this option the user needs to provide the account_no of a customer and the amount to be withdraw through user input. According to the withdraw amount the updated balance will be verified in proper format using option 5.

9. Exit the Program

The operations are choice based. Appropriate option has to be chosen from a switch-case based menu driven program and the operation on the database is performed accordingly. The output is displayed in the terminal screen with appropriate messages from the database as displayed by Oracle during direct access. Exceptions should be handled properly by the Java program. The output should be displayed in a formatted way for clarity of understanding and visual.

JAVA Program: -

BankApp.java

```
import java.sql.SQLException;
import java.util.Scanner;

public class BankApp {
   public static void main(String[] args) throws SQLException {
      System.out.println("Welcome to my Banking App\n");
      Scanner sc = new Scanner(System.in);
      while (true) {
         String menu = """
               \n\nFollowing are the available operations of the App:
               1.All Customer Records
               2.Add Customer Records
```

```
        3.Delete Customer Record
        4.Update Customer Information
        5.Account Details Of Customer
        6.Loan Details Of Customer
        7.Deposit Money To Account
        8.Withdraw Money From Account
        9.Exit
        """;
System.out.println(menu);
System.out.print("Enter operation to be performed: ");
int choice = sc.nextInt();
switch (choice) {
   case 1: {
      Customer_SQL customer = new Customer_SQL();
      customer.showCustomerRecord();
      break;
   }
   case 2: {
      Customer_SQL customer = new Customer_SQL();
      System.out.print("Enter Customer's Name: ");
      String name = sc.next();
      sc.nextLine();
      System.out.print("Enter Customer's Phone Number: ");
      String phone_no = sc.next();
      sc.nextLine();
      System.out.print("Enter Customer's City: ");
      String city = sc.next();
      sc.nextLine();
      Customer cust = new Customer(name, phone_no, city);
      customer.addCustomerRecord(cust);
      break;
   }
   case 3: {
      Customer_SQL customer = new Customer_SQL();
      customer.deleteCustomerRecord();
      break;
   }
   case 4: {
      Customer_SQL customer = new Customer_SQL();
      String menu_ = """
            1.Update Name
            2.Update Phone Number
            3.Update City
```

```
            """;
        System.out.println(menu_);
        System.out.print("Enter operation to be performed: ");
        int choice_ = sc.nextInt();
        ;
        switch (choice_) {
            case 1: {
                customer.updateName();
                break;
            }
            case 2: {
                customer.updatePhoneNumber();
                break;
            }
            case 3: {
                customer.updateCity();
                break;
            }
        }
        break;
    }
    case 5: {
        Account_SQL account = new Account_SQL();
        account.showAccountDetails();
        break;
    }
    case 6: {
        Loan_SQL loan = new Loan_SQL();
        loan.showLoanDetails();
        break;
    }
    case 7: {
        Transaction transaction = new Transaction();
        transaction.deposit();
        break;
    }
    case 8: {
        Transaction transaction = new Transaction();
        transaction.withdraw();
        break;
    }
    case 9: {
        DBConnection.getConnection().close();
```

```
                System.exit(0);
            }
        }
    }
}
```

Account_SQL.java

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class Account_SQL {
    private final Connection connection;
    public Account_SQL() throws SQLException {
        connection = DBConnection.getConnection();
    }

    public void showAccountDetails() {
        try {
            String query = """
                    SELECT
                    c.CUST_NO, c.NAME, c.PHONENO, c.CITY,
                    a.ACCOUNT_NO, a.TYPE, a.BALANCE,
                    b.BRANCH_CODE, b.BRANCH_NAME, b.BRANCH_CITY
                    FROM CUSTOMER_RECORD c
                    JOIN ACCOUNT_DETAILS a ON c.CUST_NO = a.CUST_NO
                    JOIN BRANCH b ON a.BRANCH_CODE = b.BRANCH_CODE
                    WHERE c.CUST_NO = ?;
                    """;
            PreparedStatement preparedStatement = connection.prepareStatement(query);
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter The Customer Number for Account Details: ");
            preparedStatement.setInt(1, sc.nextInt());
            sc.nextLine();
            ResultSet resultSet = preparedStatement.executeQuery();
            if (resultSet.next()) {
                System.out.println("Customer's Account & Branch information:-\n");
                System.out.println("Customer Number: " + resultSet.getInt(1));
                System.out.println("Name: " + resultSet.getString(2));
                System.out.println("Phone Number: " + resultSet.getString(3));
                System.out.println("City: " + resultSet.getString(4));
```

```
            System.out.println("Account Number: " + resultSet.getInt(5));
            System.out.println("Account Type: " + resultSet.getString(6));
            System.out.println("Account Balance: " + resultSet.getDouble(7));
            System.out.println("Branch Code: " + resultSet.getString(8));
            System.out.println("Branch Name: " + resultSet.getString(9));
            System.out.println("Branch City: " + resultSet.getString(10));
         }
         else
            System.out.println("No Such Customer Exist!!");
         sc.close();
      }
      catch (SQLException e) {
         System.out.println(e.getMessage());
      }
   }
}
```

Customer_SQL.java

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class Customer_SQL {
   private final Connection connection;

   public Customer_SQL() throws SQLException {
      connection = DBConnection.getConnection();
   }

   public void showCustomerRecord() {
      try {
         String query = "SELECT * FROM customer_record";
         Statement statement = connection.createStatement();
         ResultSet resultSet = statement.executeQuery(query);
         while (resultSet.next()) {
            System.out.println("\n\nCustomer Number :" + resultSet.getInt(1));
            System.out.println("Name :" + resultSet.getString(2));
            System.out.println("Phone Number :" + resultSet.getString(3));
            System.out.println("City :" + resultSet.getString(4));
         }
```

```
            connection.close();
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public void addCustomerRecord(Customer customer) {
        try {
            String query = "INSERT INTO customer_record (cust_no, name, phoneno, city) VALUES(?, ?, ?, ?)";
            PreparedStatement preparedStatement = connection.prepareStatement(query);
            String custQuery = "SELECT MAX(cust_no) FROM CUSTOMER_RECORD";
            Statement statement = connection.createStatement();
            ResultSet resultSet = statement.executeQuery(custQuery);
            int customer_no = 1000;
            if (resultSet.next()) {
                customer_no = resultSet.getInt(1) + 1;
            }
            customer.setCustomerNo(customer_no);
            preparedStatement.setInt(1, customer_no);
            Scanner sc = new Scanner(System.in);
            preparedStatement.setString(2, customer.getName());
            preparedStatement.setString(3, customer.getPhoneNo());
            preparedStatement.setString(4, customer.getCity());
            int rows_affected = preparedStatement.executeUpdate();
            if (rows_affected > 0) {
                System.out.println("Record Successfully Created!!");
            } else
                System.out.println("Record Creation Unsuccessful, There was an error!!");
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
        this.showCustomerRecord();
    }

    public void deleteCustomerRecord() {
        try {
            String query = "DELETE FROM customer_record WHERE cust_no = ?";
            PreparedStatement preparedStatement = connection.prepareStatement(query);
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter Customer Number of the record to be deleted: ");
            preparedStatement.setInt(1, sc.nextInt());
            sc.nextLine();
            int rows_affected = preparedStatement.executeUpdate();
```

```
        if (rows_affected > 0) {
            System.out.println("Record Successfully Deleted!!");
        } else
            System.out.println("Record Deletion Unsuccessful, There was an error!!");
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
    this.showCustomerRecord();
}

public void updateName() {
    try {
        String query = "UPDATE customer_record SET name = ? WHERE cust_no = ?";
        PreparedStatement preparedStatement = connection.prepareStatement(query);
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Customer Number: ");
        preparedStatement.setInt(2, sc.nextInt());
        sc.nextLine();
        System.out.print("Enter the name(update): ");
        preparedStatement.setString(1, sc.nextLine());
        int rows_affected = preparedStatement.executeUpdate();
        if (rows_affected > 0) {
            System.out.println("Record Updated Deleted!!");
        } else
            System.out.println("Record not updated, There was an error!!");
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
    this.showCustomerRecord();
}

public void updatePhoneNumber() {
    try {
        String query = "UPDATE customer_record SET phoneno = ? WHERE cust_no = ?";
        PreparedStatement preparedStatement = connection.prepareStatement(query);
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Customer Number: ");
        preparedStatement.setInt(2, sc.nextInt());
        sc.nextLine();
        System.out.print("Enter Phone Number(update): ");
        preparedStatement.setString(1, sc.nextLine());
        int rows_affected = preparedStatement.executeUpdate();
        if (rows_affected > 0) {
```

```
            System.out.println("Record Updated Deleted!!");
        } else
            System.out.println("Record not updated, There was an error!!");
      } catch (SQLException e) {
        System.out.println(e.getMessage());
      }
      this.showCustomerRecord();
    }

    public void updateCity() {
      try {
        String query = "UPDATE customer_record SET city = ? WHERE cust_no = ?";
        PreparedStatement preparedStatement = connection.prepareStatement(query);
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Customer Number: ");
        preparedStatement.setInt(2, sc.nextInt());
        sc.nextLine();
        System.out.print("Enter City(update): ");
        preparedStatement.setString(1, sc.nextLine());
        int rows_affected = preparedStatement.executeUpdate();
        if (rows_affected > 0) {
          System.out.println("Record Updated Deleted!!");
        } else
          System.out.println("Record not updated, There was an error!!");
      } catch (SQLException e) {
        System.out.println(e.getMessage());
      }
      this.showCustomerRecord();
    }
}
```

Customer.java

```
public class Customer {
    private int customerNo;
    private String name;
    private String phoneNo;
    private String city;

    public Customer(String name, String phoneNo, String city) {
      this.name = name;
      this.phoneNo = phoneNo;
      this.city = city;
    }
```

```
    public void setCustomerNo(int customerNo) {
        this.customerNo = customerNo;
    }

    public int getCustomerNo() {
        return this.customerNo;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return this.name;
    }

    public void setPhoneNo(String phoneNo) {
        this.phoneNo = phoneNo;
    }

    public String getPhoneNo() {
        return this.phoneNo;
    }

    public void setCity(String city) {
        this.city = city;
    }

    public String getCity() {
        return this.city;
    }
}
```

DBConnection.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {
    private static Connection connection;

    public static Connection getConnection() throws SQLException {
```

```
        if (connection == null || connection.isClosed()) {
            final String url = "jdbc:mysql://localhost:3306/banking_database";
            final String userName = "ITER_user";
            final String password = "Iter@123";
            try {
                Class.forName("com.mysql.cj.jdbc.Driver");
            } catch (ClassNotFoundException e) {
                System.out.println(e.getMessage());
            }
            try {
                connection = DriverManager.getConnection(url, userName, password);
            } catch (SQLException e) {
                System.out.println(e.getMessage());
            }
        }
        return connection;
    }
}
```

Test Cases: -

1)

```
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
1

C0001           RAJ ANAND SINGH 9861258466      DELHI
C0002           ANKITA SINGH    9879958651      BANGALORE
C0003           SOUMYA JHA      9885623344      MUMBAI
C0004           ABHIJIT MISHRA  9455845425      MUMBAI
C0005           YASH SARAF      9665854585      KOLKATA
C0006           SWAROOP RAY     9437855466      CHENNAI
C0007           SURYA NARAYAN   9937955212      GURGAON
C0008           PRANAV PRAVEEN  9336652441      PUNE
C0009           STUTI MISRA     7870266534      DELHI
C0010           ASLESHA TIWARI  null    MUMBAI
The number of rows selected is 10
```

2)

```
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
2
Enter cust_no:
C0011
Enter the name:
ANWESHA DAS
Enter the phone:
9999999999
Enter the city:
BHUB
1Row Inserted
```

```
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
2
Enter cust_no:
C0012
Enter the name:
SACHIN SINGH
Enter the phone:
9898989898
Enter the city:
CTC
1Row Inserted
```

```
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
2
Enter cust_no:
C0013
Enter the name:
ARJUN MISHRA
Enter the phone:
7777777777
Enter the city:
BBSR
1Row Inserted
```

```
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
1

C0001           RAJ ANAND SINGH 9861258466      DELHI
C0002           ANKITA SINGH    9879958651      BANGALORE
C0003           SOUMYA JHA      9885623344      MUMBAI
C0004           ABHIJIT MISHRA  9455845425      MUMBAI
C0005           YASH SARAF      9665854585      KOLKATA
C0006           SWAROOP RAY     9437855466      CHENNAI
C0007           SURYA NARAYAN   9937955212      GURGAON
C0008           PRANAV PRAVEEN  9336652441      PUNE
C0009           STUTI MISRA     7870266534      DELHI
C0010           ASLESHA TIWARI  null    MUMBAI
C0011           ANWESHA DAS     9999999999      BHUB
C0012           SACHIN SINGH    9898989898      CTC
C0013           ARJUN MISHRA    7777777777      BBSR
The number of rows selected is 13
```

3)

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
3
Enter a cust_no for deletion:
C0013
1Row deleted
```

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
3
Enter a cust_no for deletion:
C0016
0Row deleted
```

4)

```
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Num
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
4
Enter 1: For Name 2: For Phone no 3: For City to update:
2
Enter Cust No
C0011
Enter updated Phone
6370105068
1Updated Successfully
```

```
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Numb
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
4
Enter 1: For Name 2: For Phone no 3: For City to update:
1
Enter Cust No
C0011
Enter updated name
BIKASH
1Updated Successfully
```

```
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Nu
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
4
Enter 1: For Name 2: For Phone no 3: For City to update:
3
Enter Cust No
C0011
Enter updated City
ROURKELA
1Updated Successfully
```

5)

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
5
Enter Cust No
c0003
Acc. No.:       Type:   Balance:        Branch Code:    Branch Name:            Branch City:
A0001           SB       200000          B003            JUHU BRANCH     MUMBAI
Enter your choice(1-9):
5
Enter Cust No
c0016
Acc. No.:       Type:   Balance:        Branch Code:    Branch Name:            Branch City:
```

6)

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except C
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
6
Enter Cust No
c0003
SQLServerResultSet:1 no of rows affected
```

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
6
Enter Cust No
c0005
SQLServerResultSet:2 no of rows affected
C0005 YASH SARAF 9665854585 KOLKATA L0001 3000000 B006 SRIRAMPURAM BRANCH CHENNAI
```

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
6
Enter Cust No
c0008
SQLServerResultSet:3 no of rows affected
C0008 PRANAV PRAVEEN 9336652441 PUNE L0006 25000 B006 SRIRAMPURAM BRANCH CHENNAI
```

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer N
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
6
Enter Cust No
c0016
SQLServerResultSet:4 no of rows affected
```

7)

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
7
Enter Account No
a0008
Enter the amount to deposite
800
1Balance Updated
```

```
Enter your choice(1-9):
7
Enter Account No
a0005
Enter the amount to deposite
10000
1Balance Updated
```

8)

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
8
Enter Account No
a0008
Enter the amount to deposite
800
1Balance Updated
```

```
Enter your choice(1-9):
8
Enter Account No
a0008
Enter the amount to deposit
8000
Insuffient balance
```

```
Enter your choice(1-9):
8
Enter Account No
a0005
Enter the amount to deposit
10000
1 Balance Updated
```

9)

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
9
Visit us next time
```

10)

```
***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
10
Enter from 1 to 9 nothing else please again


***** Banking Management System*****
1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Record for any attribute except Customer Number
5. Show Account Details of a Customer
6. Show Loan Details of a Customer
7. Deposit Money to an Account
8. Withdraw Money from an Account
9. Exit the Program
Enter your choice(1-9):
```