

# **End Term Project Report**

## **On**

# **Introduction to Databases (CSE 3151)**

**Submitted by**

**Name : CHANDAN KUMAR MALLICK**  
**Reg. No. : 2141013059**  
**Branch : CSE**  
**Semester : 6<sup>th</sup>**  
**Section : I**  
**Session : 2023-2024**  
**Admission Batch : 2021**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**FACULTY OF ENGINEERING & TECHNOLOGY (ITER)**  
**SIKSHA 'O' ANUSANDHAN DEEMED TO BE UNIVERSITY**  
**BHUBANESWAR, ODISHA – 751030**

## **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 practised in laboratory. The Project will integrate a Java frontend menu driven program to the backend Banking Database designed in Oracle through JDBC connectivity.

**Program:*****DBConnection.java***

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

public class DBConnection {
    private static final String URL =
"jdbc:oracle:thin:@//Jeremy:1521/orclpdb";
    private static final String USER = "SCOTT";
    private static final String PASSWORD = "TIGER";

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}
```

### ***CustomerOperations.java***

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

public class CustomerOperations {
    public void showCustomerRecords(Connection connection) {
        String query = "SELECT * FROM Customer";
        System.out.println("Customer Records: ");
        System.out.println("-----");
        System.out.println("-----");
        System.out.printf("%-11s | %-20s | %-10s | %-10s\n", "Customer_No", "Name", "Phone_No", "City");
        System.out.println("-----");
        System.out.println("-----");
        try (Statement stmt = connection.createStatement(); ResultSet
rs = stmt.executeQuery(query)) {
            while (rs.next()) {
                System.out.printf("%-11s | %-20s | %-10s | %-10s\n",
                    rs.getString("Cust_No"),
                    rs.getString("Name"),
                    rs.getString("Phone_No"),
                    rs.getString("City"));
            }
            System.out.println("-----");
            System.out.println("-----");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void addCustomerRecord(Scanner scanner, Connection
connection) {
        System.out.print("Enter Customer Number: ");
        String Cust_No = scanner.nextLine();
        System.out.print("Enter Name: ");
        String Name = scanner.nextLine();
        System.out.print("Enter Phone Number: ");
        String Phone_No = scanner.nextLine();
        System.out.print("Enter City: ");
        String City = scanner.nextLine();

        String query = "INSERT INTO Customer (Cust_No, Name, Phone_No,
City) VALUES (?, ?, ?, ?)";
        try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
            pstmt.setString(1, Cust_No);
            pstmt.setString(2, Name);
            pstmt.setString(3, Phone_No);
            pstmt.setString(4, City);
            pstmt.executeUpdate();
        }
    }
}
```

```

        System.out.println("Customer added successfully.");
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public void deleteCustomerRecord(Scanner scanner, Connection
connection) {
    System.out.print("Enter Customer Number to delete: ");
    String Cust_No = scanner.nextLine();

    String query = "DELETE FROM Customer WHERE Cust_No = ?";
    try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
        pstmt.setString(1, Cust_No);
        int rowsAffected = pstmt.executeUpdate();
        if (rowsAffected > 0) {
            System.out.println("Customer deleted successfully.");
        } else {
            System.out.println("Customer not found.");
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public void updateCustomerRecord(Scanner scanner, Connection
connection) {
    System.out.print("Enter Customer Number to update: ");
    String Cust_No = scanner.nextLine();

    System.out.println("Enter 1: For Name 2: For Phone no 3: For
City to update:");
    int choice = scanner.nextInt();
    scanner.nextLine(); // Consume newline
    String column = null;
    String newValue = null;

    switch (choice) {
        case 1:
            System.out.print("Enter new Name: ");
            newValue = scanner.nextLine();
            column = "Name";
            break;
        case 2:
            System.out.print("Enter new Phone Number: ");
            newValue = scanner.nextLine();
            column = "Phone_No";
            break;
        case 3:
            System.out.print("Enter new City: ");
            newValue = scanner.nextLine();

```

```

        column = "City";
        break;
    default:
        System.out.println("Invalid choice.");
        return;
    }

    String query = "UPDATE Customer SET " + column + " = ? WHERE
Cust_No = ?";
    try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
        pstmt.setString(1, newValue);
        pstmt.setString(2, Cust_No);
        int rowsAffected = pstmt.executeUpdate();
        if (rowsAffected > 0) {
            System.out.println("Customer information updated
successfully.");
        } else {
            System.out.println("Customer not found.");
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}

```

### ***AccountLoanOperations.java***

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

public class AccountLoanOperations {
    public void showAccountDetails(Scanner scanner, Connection
connection) {
        System.out.print("Enter Customer Number: ");
        String Cust_No = scanner.nextLine();

        String query = "SELECT a.Account_No, a.Type, a.Balance,
a.Branch_Code, " +
            "b.Branch_Name, b.Branch_City, c.Cust_No, c.Name,
c.Phone_No, c.City " +
            "FROM Account a " +
            "JOIN Depositor d ON a.Account_No = d.Account_No " +
            "JOIN Customer c ON d.Cust_No = c.Cust_No " +
            "JOIN Branch b ON a.Branch_Code = b.Branch_Code " +
            "WHERE c.Cust_No = ?";

        System.out.println("-----
-----");
        System.out.printf("%-11s | %-10s | %-4s | %-8s | %-10s | %-20s
| %-10s",

"Customer_No", "Account_No", "Type", "Balance", "Branch_Code", "Branch_Name
", "Branch_City\n");
        System.out.println("-----
-----");

        try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
            pstmt.setString(1, Cust_No);
            try (ResultSet rs = pstmt.executeQuery()) {

                while (rs.next()) {
                    System.out.printf("%-11s | %-10s | %-4s | %-7s |
%-11s | %-20s | %-10s\n",

                        rs.getString("Cust_No"),
                        rs.getString("Account_No"),
                        rs.getString("Type"),
                        rs.getDouble("Balance"),
                        rs.getString("Branch_Code"),
                        rs.getString("Branch_Name"),
                        rs.getString("Branch_City"));
                }
            }
        }
    }
}
```

```

        }
        System.out.println("-----
-----");
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public void showLoanDetails(Scanner scanner, Connection
connection) {
    System.out.print("Enter Customer Number: ");
    String Cust_No = scanner.nextLine();

    String query = "SELECT l.Loan_No, l.Amount, l.Branch_Code, " +
        "b.Branch_Name, b.Branch_City, c.Cust_No, c.Name,
c.Phone_No, c.City " +
        "FROM Loan l " +
        "JOIN Customer c ON l.Cust_No = c.Cust_No " +
        "JOIN Branch b ON l.Branch_Code = b.Branch_Code " +
        "WHERE c.Cust_No = ?";

    System.out.println("-----
-----");
    System.out.printf("%-11s | %-7s | %-8s | %-10s | %-20s | %-
10s",

"Customer_No", "Loan_No", "Amount", "Branch_Code", "Branch_Name", "Branch_C
ity\n");
    System.out.println("-----
-----");

    try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
        pstmt.setString(1, Cust_No);
        try (ResultSet rs = pstmt.executeQuery()) {
            while (rs.next()) {
                System.out.printf("%-11s | %-7s | %-8s | %-11s |
%-20s | %-10s\n",

                    rs.getString("Cust_No"),
                    rs.getString("Loan_No"),
                    rs.getDouble("Amount"),
                    rs.getString("Branch_Code"),
                    rs.getString("Branch_Name"),
                    rs.getString("Branch_City"));
            }
        }
        System.out.println("-----
-----");
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```



```

    public void depositMoney(Scanner scanner, Connection connection) {
        System.out.print("Enter Account Number: ");
        String Account_No = scanner.nextLine();
        System.out.print("Enter Amount to Deposit: ");
        double Balance = scanner.nextDouble();
        scanner.nextLine(); // Consume newline

        String query = "UPDATE Account SET Balance = Balance + ? WHERE
Account_No = ?";
        try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
            pstmt.setDouble(1, Balance);
            pstmt.setString(2, Account_No);
            int rowsAffected = pstmt.executeUpdate();
            if (rowsAffected > 0) {
                System.out.println("Deposit successful.");
            } else {
                System.out.println("Account not found.");
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void withdrawMoney(Scanner scanner, Connection connection)
    {
        System.out.print("Enter Account Number: ");
        String Account_No = scanner.nextLine();
        System.out.print("Enter Amount to Withdraw: ");
        double Balance = scanner.nextDouble();
        scanner.nextLine(); // Consume newline

        // Check if sufficient balance is available
        String checkBalanceQuery = "SELECT Balance FROM Account WHERE
Account_No = ?";
        try (PreparedStatement checkStmt =
connection.prepareStatement(checkBalanceQuery)) {
            checkStmt.setString(1, Account_No);
            try (ResultSet rs = checkStmt.executeQuery()) {
                if (rs.next() && rs.getDouble("Balance") >= Balance) {
                    String query = "UPDATE Account SET Balance =
Balance - ? WHERE Account_No = ?";
                    try (PreparedStatement pstmt =
connection.prepareStatement(query)) {
                        pstmt.setDouble(1, Balance);
                        pstmt.setString(2, Account_No);
                        int rowsAffected = pstmt.executeUpdate();
                        if (rowsAffected > 0) {
                            System.out.println("Withdrawal
successful.");
                        } else {

```

```
                System.out.println("Account not found.");
            }
        }
    } else {
        System.out.println("Insufficient balance.");
    }
}
} catch (SQLException e) {
    e.printStackTrace();
}
}
```



```

        customerOps.updateCustomerRecord(scanner,
connection);
        break;
    case 5:
        accountLoanOps.showAccountDetails(scanner,
connection);
        break;
    case 6:
        accountLoanOps.showLoanDetails(scanner,
connection);
        break;
    case 7:
        accountLoanOps.depositMoney(scanner,
connection);
        break;
    case 8:
        accountLoanOps.withdrawMoney(scanner,
connection);
        break;
    case 9:
        running = false;
        break;
    default:
        System.out.println("Invalid choice. Please try
again.");
    }
}
} catch (SQLException e) {
    e.printStackTrace();
} finally {
    try {
        if (connection != null) connection.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
}

```

## Test Cases:

The program should be able to produce correct answer or appropriate error message corresponding to

the following testcases:

### 1. Show Customer Records

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 1

Customer Records:

Customer_No	Name	Phone_No	City
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

**2. Add Customer Record: <C0011, ANWESHA DAS, 9999999999, BHUB>, <c0012, SACHIN SINGH, 9898989898, CTC>, <C0013, ARJUN MISHRA, 7777777777, BBSR>**

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 2  
Enter Customer Number: C0011  
Enter Name: ANWESHA DAS  
Enter Phone Number: 9999999999  
Enter City: BHUB  
Customer added successfully.

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 2  
Enter Customer Number: c0012  
Enter Name: SACHIN SINGH  
Enter Phone Number: 9898989898  
Enter City: CTC

java.sql.SQLIntegrityConstraintViolationException: ORA-02290: check constraint  
(SCOTT.SYS\_C008292) violated

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 2  
Enter Customer Number: C0013  
Enter Name: ARJUN MISHRA  
Enter Phone Number: 7777777777  
Enter City: BBSR  
Customer added successfully.

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 1

Customer Records:

Customer_No	Name	Phone_No	City
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
C0013	ARJUN MISHRA	7777777777	BBSR

### 3. Delete Customer Record: <C0013>, <C0016>

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 3

Enter Customer Number to delete: C0013

Customer deleted successfully.

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 3

Enter Customer Number to delete: C0016

Customer not found.

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 1

Customer Records:

Customer_No	Name	Phone_No	City
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



#### 4. Update Customer Record for any attribute except Customer Number: <C0011>

[ Update each column once]

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 4

Enter Customer Number to update: C0001

Enter 1: For Name 2: For Phone no 3: For City to update:

2

Enter new Phone Number: 7327888730

Customer information updated successfully.

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 1

Customer Records:

Customer_No	Name	Phone_No	City
C0001	RAJ ANAND SINGH	7327888730	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

---

**5. Show Account Details of a Customer: <C0003>, <c0005>, <C0016>**

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 5

Enter Customer Number: C0003

---

Customer_No	Account_No	Type	Balance	Branch_Code	Branch_Name	Branch_City
-------------	------------	------	---------	-------------	-------------	-------------

---

C0003	A0001	SB	200000.0	B003	JUHU BRANCH	MUMBAI
-------	-------	----	----------	------	-------------	--------

---

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 5

Enter Customer Number: c0005

---

Customer_No	Account_No	Type	Balance	Branch_Code	Branch_Name	Branch_City
-------------	------------	------	---------	-------------	-------------	-------------

---

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 5

Enter Customer Number: C0016

-----  
Customer\_No | Account\_No | Type | Balance | Branch\_Code | Branch\_Name | Branch\_City  
-----  
-----

### 6. Show Loan Details of a Customer: <C0003>, <c0005>, <C0008>, <C0016>

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 6

Enter Customer Number: C0003

-----  
Customer\_No | Loan\_No | Amount | Branch\_Code | Branch\_Name | Branch\_City  
-----  
-----

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 6

Enter Customer Number: c0005

-----  
Customer\_No | Loan\_No | Amount | Branch\_Code | Branch\_Name | Branch\_City  
-----  
-----

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 6

Enter Customer Number: C0008

Customer_No	Loan_No	Amount	Branch_Code	Branch_Name	Branch_City
C0008	L0006	25000.0	B006	SRIRAMPURAM BRANCH	CHENNAI

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 6

Enter Customer Number: C0016

Customer_No	Loan_No	Amount	Branch_Code	Branch_Name	Branch_City
-------------	---------	--------	-------------	-------------	-------------

### 7. Deposit Money to an Account: <A0008, 800>, <a0005, 10000>

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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  
Enter your choice (1-9): 7  
Enter Account Number: A0008  
Enter Amount to Deposit: 800  
Deposit successful.

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records  
2. Add Customer Record  
3. Delete Customer Record  
4. Update Customer Information  
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  
Enter your choice (1-9): 7  
Enter Account Number: a0005  
Enter Amount to Deposit: 10000  
Account not found.

**8. Withdraw Money from an Account: <A0008, 800>, <A0008, 8000><a0005, 10000>**

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records  
2. Add Customer Record  
3. Delete Customer Record  
4. Update Customer Information  
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  
Enter your choice (1-9): 8  
Enter Account Number: A0008  
Enter Amount to Withdraw: 800  
Withdrawal successful.

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records  
2. Add Customer Record  
3. Delete Customer Record  
4. Update Customer Information

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

Enter your choice (1-9): 8

Enter Account Number: A0008

Enter Amount to Withdraw: 8000

Insufficient balance.

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 8

Enter Account Number: a0005

Enter Amount to Withdraw: 10000

Insufficient balance.

## **9. Exit the Program**

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record
3. Delete Customer Record
4. Update Customer Information
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

Enter your choice (1-9): 9

Process finished with exit code 0

## **10. Enter choice 10**

\*\*\*\*\* Banking Management System \*\*\*\*\*

1. Show Customer Records
2. Add Customer Record

- 3. Delete Customer Record
- 4. Update Customer Information
- 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

Enter your choice (1-9): 10

Invalid choice. Please try again.