**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.

1. 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.

1. 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.
2. 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.

1. 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.

1. 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.

1. 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.

1. 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.

1. 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());

            }

        }

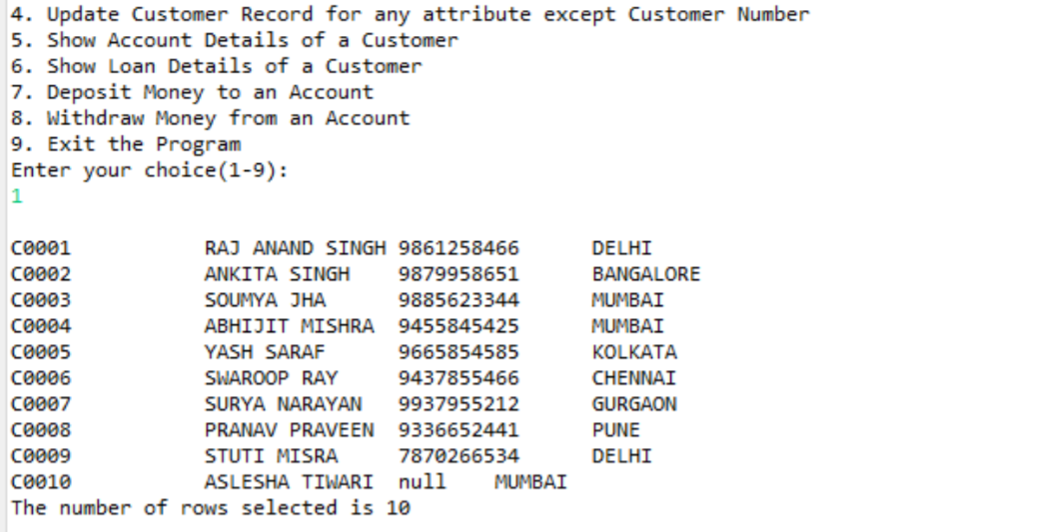
        return connection;

    }

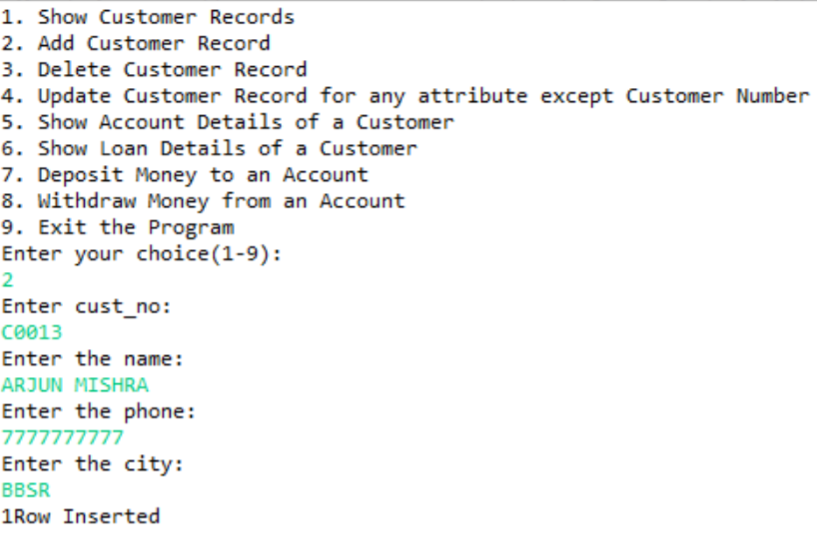
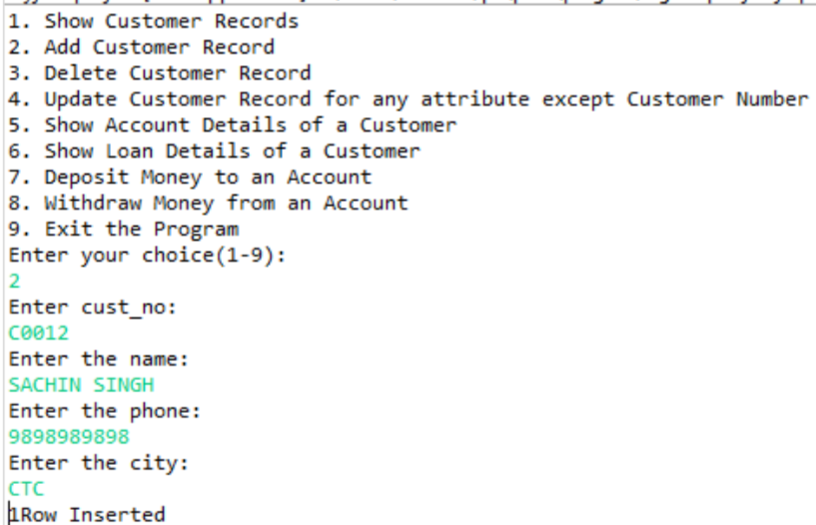
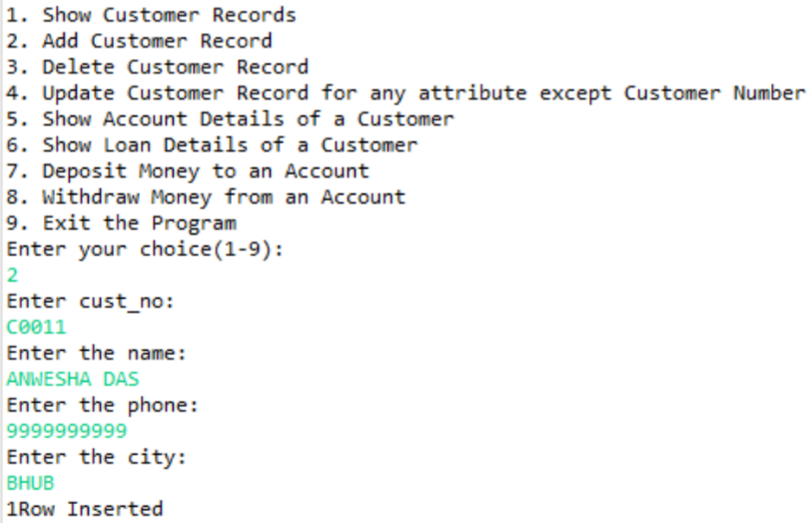
}

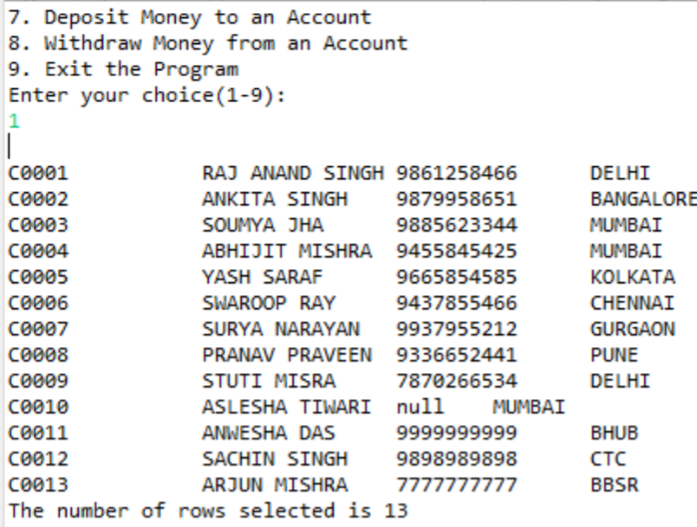
**Test Cases:** **-**

1)

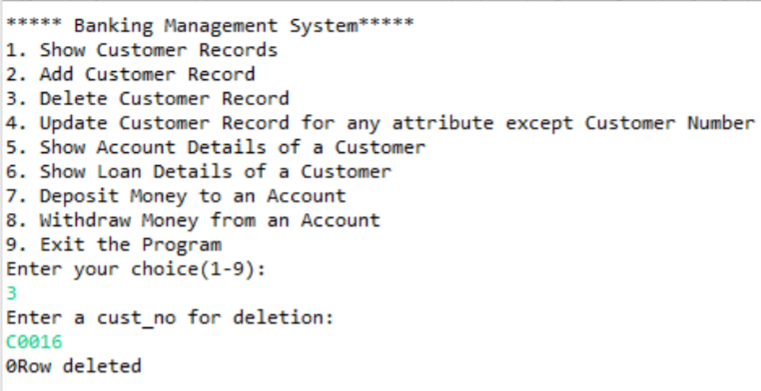
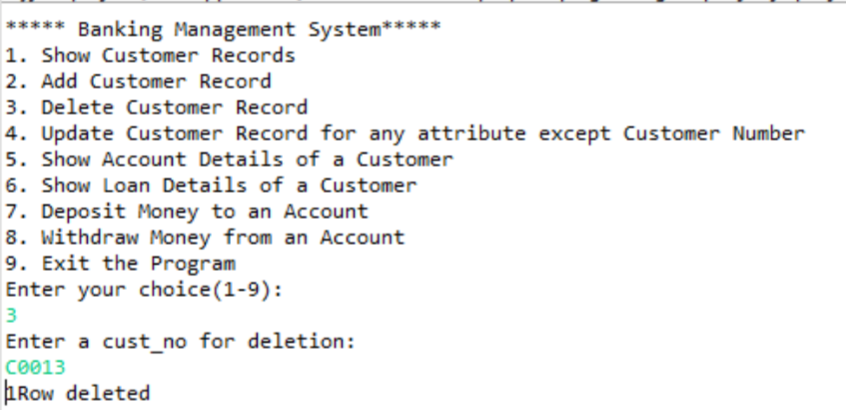


2)

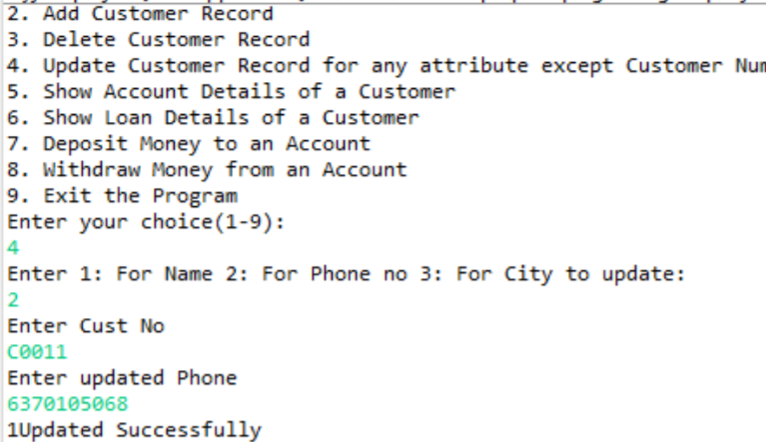
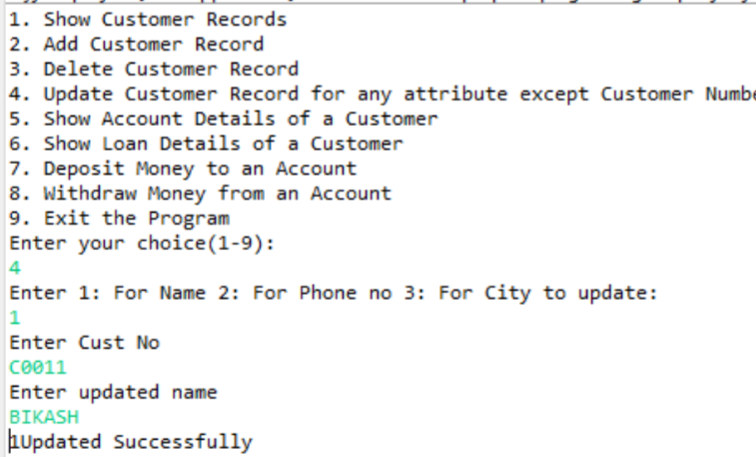
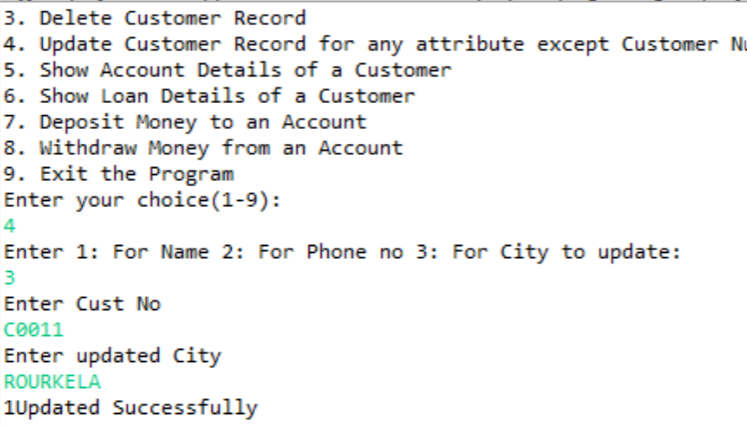




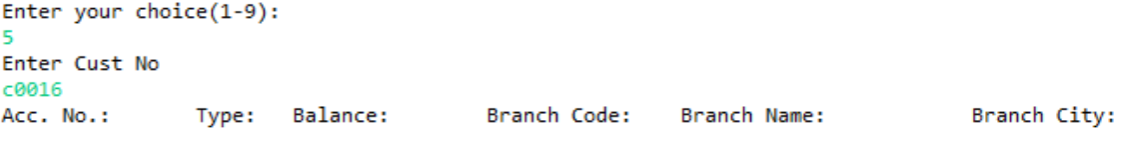
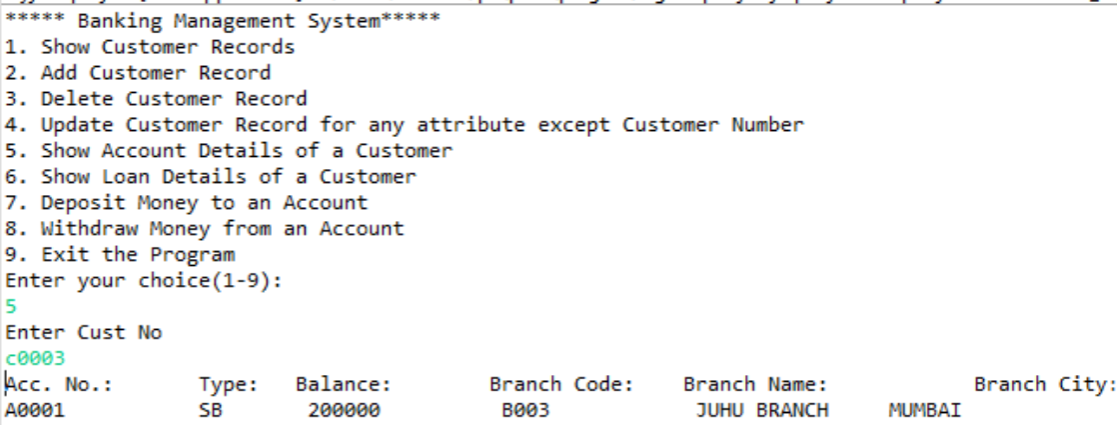
3)



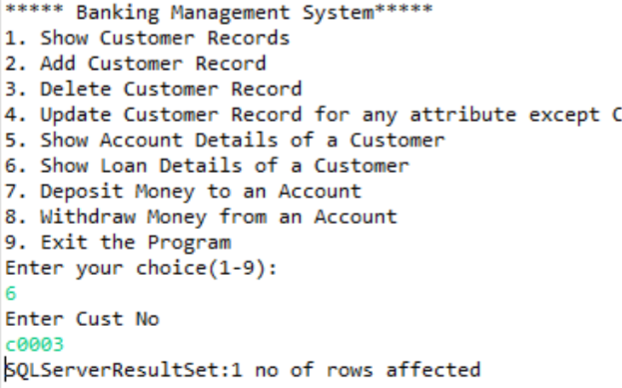
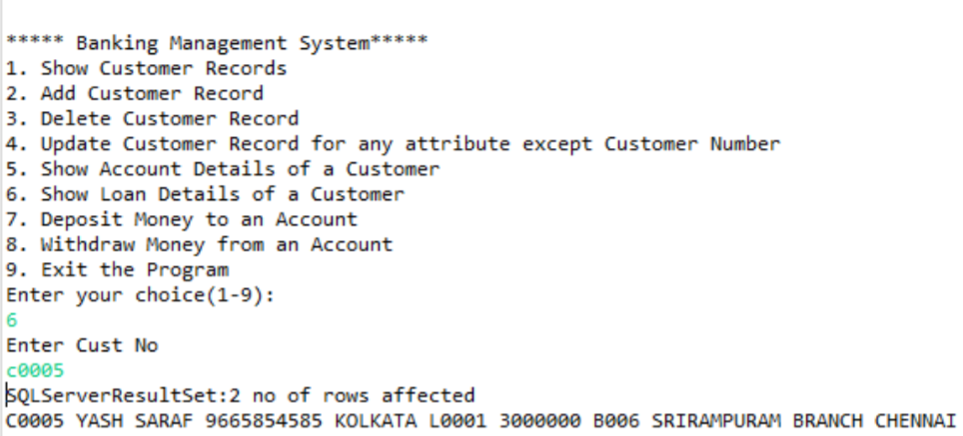
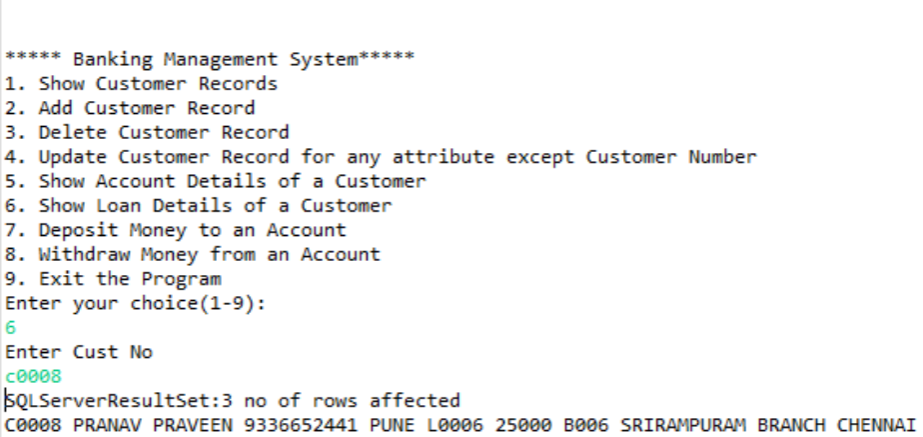
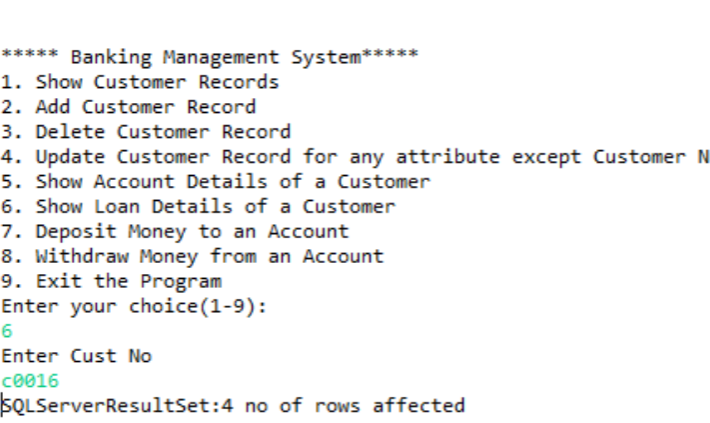
4)

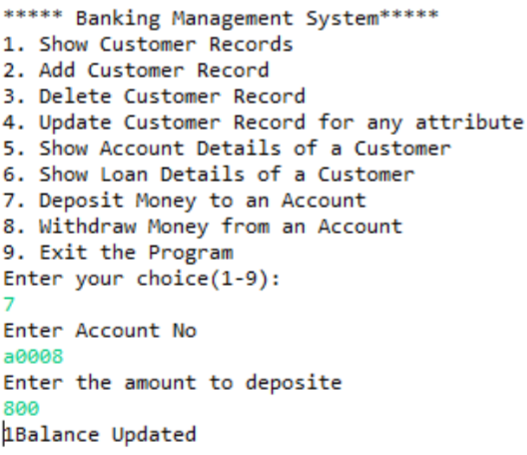
5)

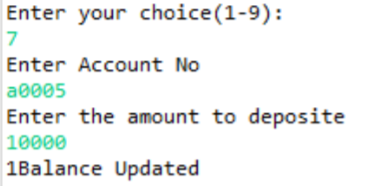


6)

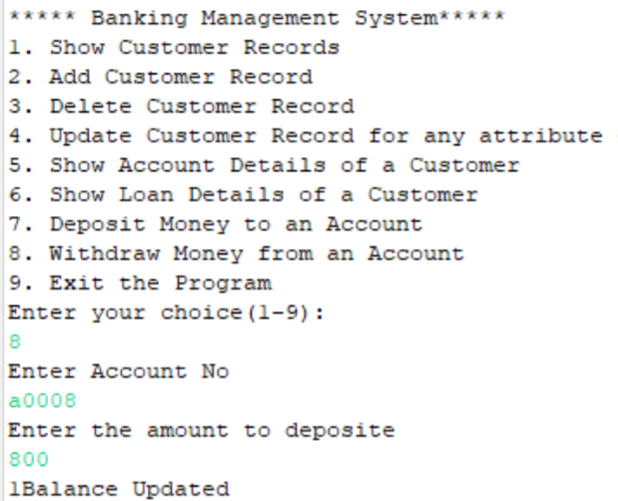
   

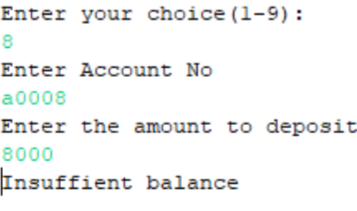
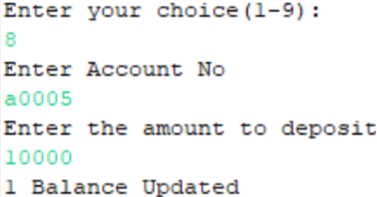
7)



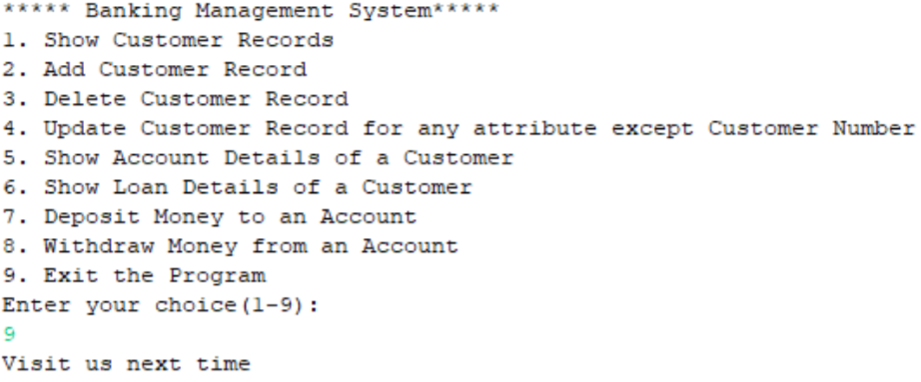


8)



9)



10)

