

End-Term Project Report
On
Introduction to Databases (CSE 3151)
[Banking Management System]

Submitted by

Name: Utsav kr. Choudhary

Registration No.: 2141003017

Branch: CSE

Semester: 6th Section: 2141005



Department of Computer Science & Engineering
Faculty of Engineering & Technology (ITER)
SIKSHA 'O' ANUSANDHAN DEEMED TO BE UNIVERSITY
BHUBANESWAR, ODISHA - 751030

Objective of this EndTerm Project:

- To design a miniature Project for a Banking Management System using Java, Oracle and JDBC.

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.

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
- 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 the Program

Test Cases:

The program should be able to produce correct answer or appropriate error message corresponding to the following testcases:

1. Show Customer Records
2. Add Customer Record:
 - <C0011, ANWESHA DAS, 9999999999, BHUB>
 - <c0012, SACHIN SINGH, 9898989898, CTC>
 - <C0013, ARJUN MISHRA, 7777777777, BBSR>
3. Delete Customer Record: <C0013>, <C0016>
4. Update Customer Record for any attribute except Customer Number: [Update each column once]
5. Show Account Details of a Customer: <C0003>, <c0005>, <C0016>
6. Show Loan Details of a Customer: <C0003>, <c0005>, <C0008>, <C0016>
7. Deposit Money to an Account: <A0008, 800>, <a0005, 10000>
8. Withdraw Money from an Account: <A0008, 800>, <A0008, 8000><a0005, 10000>
9. Exit the Program
10. Enter choice 10

```
IDEndTermProject.java ×
1 import java.sql.*;
2 import java.util.Scanner;
3
4 public class IDEndTermProject {
5     static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
6     static final String DB_URL = "jdbc:mysql://localhost/idproject1";
7
8     static final String USER = "root";
9     static final String PASS = "csee2141003017";
10
11     public static void main(String[] args) {
12         Connection conn = null;
13         Statement stmt = null;
14         try {
15             Class.forName(JDBC_DRIVER);
16             System.out.println("Connecting to database...");
17             conn = DriverManager.getConnection(DB_URL, USER, PASS);
18             System.out.println("Connected to database successfully...");
19
20             stmt = conn.createStatement();
21
22             // Menu-driven program
23             int choice;
24             Scanner scanner = new Scanner(System.in);
25             do {
26                 System.out.println("\n***** Banking Management System *****");
27                 System.out.println("1. Show Customer Records");
28                 System.out.println("2. Add Customer Record");
29                 System.out.println("3. Delete Customer Record");
30                 System.out.println("4. Update Customer Information");
31                 System.out.println("5. Show Account Details of a Customer");
32                 System.out.println("6. Show Loan Details of a Customer");
33                 System.out.println("7. Deposit Money to an Account");
34                 System.out.println("8. Withdraw Money from an Account");
35                 System.out.println("9. Exit");
36
37                 System.out.print("Enter your choice (1-9): ");
38                 choice = Integer.parseInt(scanner.nextLine());
39
40                 switch (choice) {
41                     case 1:
42                         showCustomerRecords(stmt);
43                         break;
44                     case 2:
45                         addCustomerRecord(stmt, scanner);
46                         break;
47                     case 3:
48                         deleteCustomerRecord(stmt, scanner);
49                         break;
50                     case 4:
51                         updateCustomerInformation(stmt, scanner);
52                         break;
53                     case 5:
54                         showAccountDetails(stmt, scanner);
55                         break;
56                     case 6:
57                         showLoanDetails(stmt, scanner);
58                         break;
59                     case 7:
60                         depositMoney(stmt, scanner);
61                         break;
62                     case 8:
63                         withdrawMoney(stmt, scanner);
64                         break;
65                     case 9:
66                         System.out.println("Exiting the program...");
67                         break;
68                     default:
69                         System.out.println("Invalid choice. Please enter a number between 1 and 9.");
70                 }
71             }
72         }
73     }
74 }
```

```
71         } while (choice != 9);
72
73     } catch (Exception e) {
74         e.printStackTrace();
75     } finally {
76         try {
77             if (stmt != null) stmt.close();
78             if (conn != null) conn.close();
79         } catch (SQLException se) {
80             se.printStackTrace();
81         }
82     }
83 }
84
85 private static void showCustomerRecords(Statement stmt) throws SQLException {
86     ResultSet rs = stmt.executeQuery("SELECT * FROM CUSTOMERS");
87
88     System.out.println("Customer Records:");
89     System.out.println("-----");
90     System.out.printf("%-10s | %-20s | %-15s | %-10s\n", "Customer No", "Name", "Phone", "City");
91     System.out.println("-----");
92
93     while (rs.next()) {
94         System.out.printf("%-10s | %-20s | %-15s | %-10s\n",
95             rs.getString("CUST_NO"),
96             rs.getString("NAME"),
97             rs.getString("PHONE"),
98             rs.getString("CITY"));
99     }
100    System.out.println("-----");
101    rs.close();
102 }
103
104 private static void addCustomerRecord(Statement stmt, Scanner scanner) throws SQLException {
105     System.out.print("Enter Customer Number: ");
106     String custNo = scanner.nextLine();
107     System.out.print("Enter Name: ");
108     String name = scanner.nextLine();
109     System.out.print("Enter Phone Number: ");
110     String phone = scanner.nextLine();
111     System.out.print("Enter City: ");
112     String city = scanner.nextLine();
113
114     String sql = "INSERT INTO CUSTOMERS (CUST_NO, NAME, PHONE, CITY) VALUES (?, ?, ?, ?)";
115     PreparedStatement pstmt = stmt.getConnection().prepareStatement(sql);
116     pstmt.setString(1, custNo);
117     pstmt.setString(2, name);
118     pstmt.setString(3, phone);
119     pstmt.setString(4, city);
120     int rowsAffected = pstmt.executeUpdate();
121     if (rowsAffected > 0) {
122         System.out.println("Customer added successfully.");
123     } else {
124         System.out.println("Failed to add customer.");
125     }
126     pstmt.close();
127 }
128
129 private static void deleteCustomerRecord(Statement stmt, Scanner scanner) throws SQLException {
130     System.out.print("Enter Customer Number to delete: ");
131     String custNo = scanner.nextLine();
132
133     String sql = "DELETE FROM CUSTOMERS WHERE CUST_NO = ?";
134     PreparedStatement pstmt = stmt.getConnection().prepareStatement(sql);
135     pstmt.setString(1, custNo);
136     int rowsAffected = pstmt.executeUpdate();
137     if (rowsAffected > 0) {
138         System.out.println("Customer deleted successfully.");
139     }
```

```
139     } else {  
140         System.out.println("Customer not found or deletion failed.");  
141     }  
142     pstmt.close();  
143 }  
144  
145 private static void updateCustomerInformation(Statement stmt, Scanner scanner) throws SQLException {  
146     System.out.print("Enter Customer Number to update: ");  
147     String custNo = scanner.nextLine();  
148     System.out.println("Choose attribute to update:");  
149     System.out.println("1. Name");  
150     System.out.println("2. Phone Number");  
151     System.out.println("3. City");  
152     System.out.print("Enter your choice (1-3): ");  
153     int attributeChoice = Integer.parseInt(scanner.nextLine());  
154  
155     String attributeToUpdate = null;  
156     switch (attributeChoice) {  
157         case 1:  
158             attributeToUpdate = "NAME";  
159             break;  
160         case 2:  
161             attributeToUpdate = "PHONE";  
162             break;  
163         case 3:  
164             attributeToUpdate = "CITY";  
165             break;  
166         default:  
167             System.out.println("Invalid choice.");  
168             return;  
169     }  
170  
171     System.out.print("Enter new value for " + attributeToUpdate + ": ");  
172     String newValue = scanner.nextLine();  
173  
174     String sql = "UPDATE CUSTOMERS SET " + attributeToUpdate + " = ? WHERE CUST_NO = ?";  
175     PreparedStatement pstmt = stmt.getConnection().prepareStatement(sql);  
176     pstmt.setString(1, newValue);  
177     pstmt.setString(2, custNo);  
178     int rowsAffected = pstmt.executeUpdate();  
179     if (rowsAffected > 0) {  
180         System.out.println("Customer information updated successfully.");  
181     } else {  
182         System.out.println("Customer not found or update failed.");  
183     }  
184     pstmt.close();  
185 }  
186  
187 private static void showAccountDetails(Statement stmt, Scanner scanner) throws SQLException {  
188     System.out.print("Enter Customer Number: ");  
189     String custNo = scanner.nextLine();  
190  
191     String sql = "SELECT * FROM ACCOUNTS WHERE CUST_NO = ?";  
192     PreparedStatement pstmt = stmt.getConnection().prepareStatement(sql);  
193     pstmt.setString(1, custNo);  
194     ResultSet rs = pstmt.executeQuery();  
195  
196     System.out.println("-----");  
197     System.out.println("Account Number | Type | Balance | Branch Code | Branch Name | Branch City");  
198     System.out.println("-----");  
199     while (rs.next()) {  
200         System.out.printf("%-15s | %-4s | %-8s | %-11s | %-12s | %-11s\n",  
201             rs.getString("ACCOUNT_NO"),  
202             rs.getString("TYPE"),  
203             rs.getString("BALANCE"),  
204             rs.getString("BRANCH_CODE"),  
205             rs.getString("BRANCH_NAME"),  
206             rs.getString("BRANCH_CITY"));  
207     }  
208     System.out.println("-----");  
209  
210     rs.close();  
211     pstmt.close();  
212 }  
213  
214
```

```
215 private static void showLoanDetails(Statement stmt, Scanner scanner) throws SQLException {
216     System.out.print("Enter Customer Number: ");
217     String custNo = scanner.nextLine();
218
219     String sql = "SELECT * FROM LOANS WHERE CUST_NO = ?";
220     PreparedStatement pstmt = stmt.getConnection().prepareStatement(sql);
221     pstmt.setString(1, custNo);
222     ResultSet rs = pstmt.executeQuery();
223
224     System.out.println("-----");
225     System.out.println("Loan Number | Loan Amount | Branch Code | Branch Name | Branch City");
226     System.out.println("-----");
227     while (rs.next()) {
228         System.out.printf("%-11s | %-11s | %-11s | %-12s | %-11s\n",
229             rs.getString("LOAN_NO"),
230             rs.getString("LOAN_AMOUNT"),
231             rs.getString("BRANCH_CODE"),
232             rs.getString("BRANCH_NAME"),
233             rs.getString("BRANCH_CITY"));
234     }
235     System.out.println("-----");
236
237     rs.close();
238     pstmt.close();
239 }
240
241
242 private static void depositMoney(Statement stmt, Scanner scanner) throws SQLException {
243     System.out.print("Enter Account Number: ");
244     String accountNo = scanner.nextLine();
245     System.out.print("Enter Amount to Deposit: ");
246     double amount = Double.parseDouble(scanner.nextLine());
247
248     String sql = "UPDATE ACCOUNTS SET BALANCE = BALANCE + ? WHERE ACCOUNT_NO = ?";
249     PreparedStatement pstmt = stmt.getConnection().prepareStatement(sql);
250     pstmt.setDouble(1, amount);
251     pstmt.setString(2, accountNo);
252     int rowsAffected = pstmt.executeUpdate();
253     if (rowsAffected > 0) {
254         System.out.println("Money deposited successfully.");
255     } else {
256         System.out.println("Failed to deposit money.");
257     }
258     pstmt.close();
259 }
260
261 private static void withdrawMoney(Statement stmt, Scanner scanner) throws SQLException {
262     System.out.print("Enter Account Number: ");
263     String accountNo = scanner.nextLine();
264     System.out.print("Enter Amount to Withdraw: ");
265     double amount = Double.parseDouble(scanner.nextLine());
266
267     String sql = "SELECT BALANCE FROM ACCOUNTS WHERE ACCOUNT_NO = ?";
268     PreparedStatement balanceStmt = stmt.getConnection().prepareStatement(sql);
269     balanceStmt.setString(1, accountNo);
270     ResultSet rs = balanceStmt.executeQuery();
271     if (rs.next()) {
272         double currentBalance = rs.getDouble("BALANCE");
273         if (currentBalance >= amount) {
274             sql = "UPDATE ACCOUNTS SET BALANCE = BALANCE - ? WHERE ACCOUNT_NO = ?";
275             PreparedStatement withdrawStmt = stmt.getConnection().prepareStatement(sql);
276             withdrawStmt.setDouble(1, amount);
277             withdrawStmt.setString(2, accountNo);
278             int rowsAffected = withdrawStmt.executeUpdate();
279             if (rowsAffected > 0) {
280                 System.out.println("Money withdrawn successfully.");
281             } else {
282                 System.out.println("Failed to withdraw money.");
283             }
284             withdrawStmt.close();
285         }
286     }
287 }
```



```
285         } else {  
286             System.out.println("Insufficient balance.");  
287         }  
288     } else {  
289         System.out.println("Account not found.");  
290     }  
291     rs.close();  
292     balanceStmt.close();  
293 }  
294 }  
295
```

1. Show Customer Records

Enter your choice (1-9): 1

Customer Records:

Customer No	Name	Phone	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>

Enter your choice (1-9): 2

Enter Customer Number: C0011

Enter Name: ANWESHA DAS

Enter Phone Number: 9999999999

Enter City: BHUB

Customer added successfully.

Enter your choice (1-9): 2

Enter Customer Number: C0012

Enter Name: SACHIN SINGH

Enter Phone Number: 9898989898

Enter City: CTC

Customer added successfully.

Enter your choice (1-9): 2

Enter Customer Number: C0013

Enter Name: ARJUN MISHRA

Enter Phone Number: 7777777777

Enter City: BBSR

Customer added successfully.

OUTPUT:

```
Enter your choice (1-9): 1
```

```
Customer Records:
```

```
-----  
Customer No | Name                | Phone          | 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  
C0012      | SACHIN SINGH        | 9898989898    | CTC  
C0013      | ARJUN MISHRA        | 7777777777    | BBSR  
-----
```

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

```
Enter your choice (1-9): 3
```

```
Enter Customer Number to delete: C0013
```

```
Customer deleted successfully.
```

C0016-0ROW DELETED

4. Update Customer Record for any attribute except Customer Number: [Update each column once]

```
Enter your choice (1-9): 4
```

```
Enter Customer Number to update: C0011
```

```
Choose attribute to update:
```

1. Name
2. Phone Number
3. City

```
Enter your choice (1-3): 1
```

```
Enter new value for NAME: THALA FOR A REASON
```

```
Customer information updated successfully.
```

```
Enter your choice (1-9): 4
```

```
Enter Customer Number to update: C0011
```

```
Choose attribute to update:
```

1. Name
2. Phone Number
3. City

```
Enter your choice (1-3): 2
```

```
Enter new value for PHONE: 0707070707
```

```
Customer information updated successfully.
```



```
Enter your choice (1-9): 4
Enter Customer Number to update: C0011
Choose attribute to update:
1. Name
2. Phone Number
3. City
Enter your choice (1-3): 3
Enter new value for CITY: RANCHI
Customer information updated successfully.
```

OUTPUT:

```
Enter your choice (1-9): 1
Customer Records:
-----
Customer No | Name                | Phone          | 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       | THALA FOR A REASON  | 0707070707    | RANCHI
C0012       | SACHIN SINGH        | 9898989898    | CTC
-----
```

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

```
Enter your choice (1-9): 5
Enter Customer Number: C0003
-----
Account Number | Type | Balance | Branch Code | Branch Name | Branch City
-----
A0003          | CA   | 850000.00 | B004        | ANDHERI BRANCH | MUMBAI
-----
```

```
Enter your choice (1-9): 5
Enter Customer Number: C0005
-----
Account Number | Type | Balance | Branch Code | Branch Name | Branch City
-----
A0005          | FD   | 28500.00 | B005        | SALT LAKE BRANCH | KOLKATA
-----
```

```
Enter your choice (1-9): 5
Enter Customer Number: C0016
```

```
-----
Account Number | Type | Balance | Branch Code | Branch Name | Branch City
-----
-----
```

6. Show Loan Details of a Customer: <C0003>, <C0005>, <C0008>, <C0016>

```
Enter your choice (1-9): 6
Enter Customer Number: C0003
```

```
-----
Loan Number | Loan Amount | Branch Code | Branch Name | Branch City
-----
-----
```

```
Enter your choice (1-9): 6
Enter Customer Number: C0005
```

```
-----
Loan Number | Loan Amount | Branch Code | Branch Name | Branch City
-----
L0001      | 3000000.00 | B006      | SHRIRAMPURAM BRANCH | CHENNAI
-----
```

```
Enter your choice (1-9): 6
Enter Customer Number: C0008
```

```
-----
Loan Number | Loan Amount | Branch Code | Branch Name | Branch City
-----
L0006      | 25000.00   | B006      | SHRIRAMPURAM BRANCH | CHENNAI
-----
```

```
Enter your choice (1-9): 6
Enter Customer Number: C0016
```

```
-----
Loan Number | Loan Amount | Branch Code | Branch Name | Branch City
-----
-----
```

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

```
Enter your choice (1-9): 7
Enter Account Number: A0008
Enter Amount to Deposit: 800
Money deposited successfully.
```

```
Enter your choice (1-9): 7
Enter Account Number: A0005
Enter Amount to Deposit: 10000
Money deposited successfully.
```

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

```
Enter your choice (1-9): 8
Enter Account Number: A0008
Enter Amount to Withdraw: 800
Money withdrawn successfully.
```

```
Enter your choice (1-9): 8
Enter Account Number: A0008
Enter Amount to Withdraw: 8000
Insufficient balance.
```

```
Enter your choice (1-9): 8
Enter Account Number: A0005
Enter Amount to Withdraw: 10000
Money withdrawn successfully.
```

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
Exiting the program...
```