# 

# LIBRARY MANAGEMENT SYSTEM

**Documentation of Case Study**

**Member Details:**

|  |  |
| --- | --- |
| **Employee ID** | **Employee Name** |
| 139614 | Pavithra |
| 139610 | Vipul Sinha |
| 139613 | Carolin James |
| 139606 | Pawan Yadav |
| 139626 | Nomaan Imtiaz |

## **Description of Different coding blocks:**

**Library Package-**

1. **AddBooks:**

This functionality is used to add a book to the library. .In which a class **addBook** is created with data members such as **Publication**, **Bookname**, **Author**, **Copies\_Available** with their respective datatypes. Then a **try** block which fetches database table using JDBC connection and to connect with the specified table and package and to call the stored procedures callable statement is used. The user inputs are taken and set to the to the respective Data members and at last a catch block if any exception arises while connection the JDBC and then connections and the callable statements are closed.

1. **AddBookLocation:**

This functionality is used to consist the location of the book. In which a class **addBookLocations** is created with the data members such as **Floor\_no,Cupboard\_no,rack\_no,book\_id** with their respective datatypes. Then a **try** block which fetches database table using JDBC connection and to connect with the specified table and package and to call the stored procedures callable statement is used. The user inputs are taken and set to the to the respective Data members and at last a catch block if any exception arises while connection the JDBC and then connections and the callable statements are closed.

1. **AddNewMember:**

This functionality is used to add member in which it consist of complete detail of the member.In which a class add Member is created with data member such as **First\_name,Last\_name,Password,Mobile,Address.**Then a try block which fetches database table using JDBC connection and to connect with the specified table and package and to call the stored procedures callable statement is used. The user inputs are taken and at last a catch block if any exception arises while connection the JDBC and then connections and the callable statements are closed using the finally block.

1. **ReturnBooks:**

This functionality is used to return a book to the library In which a class **Return Book** . with data member such as member\_id,book\_id is used. Then a **try** block which fetches database table using JDBC connection and to connect with the specified table and package and to call the stored procedures callable statement is used. The user inputs are taken and at last a catch block if any exception arises while connection the JDBC and then connections and the callable statements are closed.

1. **SearchBookLocation:**

This functionality is used to find the book location with their shelf number and rack number.In which a class **SearchBookLocation** is created.Then the book name as string is given as a user input and it return the books location with the attribute floor\_no,cupboard\_no,rack\_no. Then a **try** block which fetches database table using JDBC connection and to connect with the specified table and package and to call the stored procedures callable statement is used. The user inputs are taken and at last a catch block if any exception arises while connection the JDBC and then connections and the callable statements are closed.

1. **DeleteBookLocations:**

This functionality is used to delete the book location.In which a class **DeleteBookLocations** is created.Then the book\_id as Integer is given as a user input and it deletes the books location . Then a try block which fetches database table using JDBC connection and to connect with the specified table and package and to call the stored procedures callable statement is used. The user inputs are taken and at last a catch block if any exception arises while connection the JDBC and then connections and the callable statements are closed.

1. **SearchMemberByName:**

This functionality is used to search member and display his/her details using first\_name or last\_name. In which a class SearchMemberByName is created with data member such as First\_name,Last\_name and used as input.Then a **try** block which fetches database table using JDBC connection and to connect with the specified table and package and to call the stored procedures callable statement is used. The user inputs are taken and at last a catch block if any exception arises while connection the JDBC and then connections and the callable statements are closed.

1. **SearchMemberById:**

This functionality is to find whether the member is part if the database or not.

It is used to search member and display his/her details using member\_id. In which a class SearchMemberById is created with data member such as Member\_id and used as input.Then a try block which fetches database table using JDBC connection and to connect with the specified table and package and to call the stored procedures callable statement is used. The user inputs are taken and at last a catchblock if any exception arises while connection the JDBC and then connections and the callable statements are closed.

1. **SearchByBookId:**

This functionality is to search and display details of book by using book\_id.

In which a class SearchByBookId is created with data member such as Book\_id and used as input.Then a try block which fetches database table using JDBC connection and to connect with the specified table and package and to call the stored procedures callable statement is used. The user inputs are taken and at last a catchblock if any exception arises while connection the JDBC and then connections and the callable statements are closed.

1. **BookIssue:**

The functionality of this class is to Issue book by takin member id and book id from the librarian as the input. The rest of the details like issue date, expected return date won’t be asked from the user. It will be automatically generated by the system.

With the use of Statement we are creating a query to get the issue id of the book returned. After returning the book the class will check whether the return date is after the expected return date or not. If the return date is greater than the expected return date then the receipt will be generated for the user in which details of the fine will be showed along with the receipt id, book name, issue date, return date, name of member.

1. **ViewIssueBookRecord :**

The functionality of this class is to show all the member who have issued a particular book which is not returned yet and the book id will be entered by the librarian. This class will help the librarian to find out the books which is not returned by the users.

**Member Package-**

**1. ViewIssuedBook :**

The functionality of this class is to show the book name, issue date and expected return date of the books which is currently issued by him. It will help the user to track the return dates of the books he has issued in order to avoid fine. Use of callable statement has been used to call stored PLSQL procedure.

**2. ViewHistory :**

The class will show the book issuing history of the particular member when called.

This class will be called from the driver class passing parameter that is member id into it method. It will then display the list of all the books which were issued by the member previously and has been returned.

**3. ViewPopularBook :**

The function will show the most popular books of the library to the member. The more number of times a book is issued the more popularity it will have.

**4. ChangePassword :**

This class will be used to change the password of the member. It will ask the user to enter the new password and then it will ask the user to confirm password. If the new password and confirm password is same then it will change the password of the user.

**Driver Package –**

1. **LibrarianValidation :**

The functionality of this class is to validate the librarian credentials. It will ask the librarian to enter its username and password. If the username exists in the database and the password matches with it, It will give an interface to librarian which will consists of all the functionalities that is present in the library package.

1. **MemberValidation :**

The functionality of this class is to the validation of member. It first asks the member to enter his member id and password. Then it checks the database whether the user exists and the member id and password matches or not. If it matches then it gives the member an interface where all the functionalities of member package is present in the interface.

1. **LibraryManagementSystem :**

It is the driver class for all the library management system. It will first start with an interface where it will ask whether the user is a member or librarian. After enter the option it will ask the user for validation. Once the validation is completed it will give user their corresponding interfaces where they can use all the functionalities which are present in their corresponding packages.

**Jdbcconnection Package** -

**Jdbcconnection :**

This class is used for database connection. All the classes which needs to interact with the database imports this Class into their file.

**SOURCE CODE**-

**Jdbcconnection Package**-

**JdbcConnection.java**

package jdbcconnection;

import java.sql.Connection;

import java.sql.DriverManager;

public class JdbcConnection

{

private static final String regitertodatabase = "oracle.jdbc.driver.OracleDriver";

private static final String drivermanager = "jdbc:oracle:thin:@localhost:1521:xe";

private static final String username = "hr";

private static final String password ="hr";

public static Connection getConnection()

{

Connection connection = null;

try

{

Class.forName(regitertodatabase);

connection = DriverManager.getConnection(drivermanager, username, password);

if(connection ==null)

{

System.out.println("connection is unsuccessful..");

}

}

catch(Exception e)

{

System.out.println(e);

}

return connection;

}

}

**Library Package**-

**Addbook.java**

package library;

import java.sql.Connection;

import jdbcconnection.JdbcConnection;

import java.sql.CallableStatement;

import java.sql.SQLException;

import java.util.Scanner;

public class AddBook

{

static Connection connection;

static CallableStatement callablestatement = null;

public static void addBook() throws SQLException

{

String bookname;

String author;

String publication;

int copiesavailable;

Scanner scanner = new Scanner(System.in);

try

{

connection= JdbcConnection.getConnection();

callablestatement = connection.prepareCall("call library.addBooks(?,?,?,?)");

System.out.println("Enter the Book's Name : ");

bookname = scanner.next();

System.out.println("Enter the Author`s name : ");

author = scanner.next();

System.out.println("Enter the publication : ");

publication = scanner.next();

System.out.println("Enter number of copies available : ");

copiesavailable = scanner.nextInt();

callablestatement.setString(1,bookname);

callablestatement.setString(2,author);

callablestatement.setString(3,publication);

callablestatement.setInt(4,copiesavailable);

callablestatement.executeUpdate();

System.out.println("Book Inserted!");

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

connection.close();

callablestatement.close();

scanner.close();

}

}

}

**AddBookLocation .java**

package library;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.CallableStatement;

import java.util.Scanner;

import java.sql.SQLException;

import jdbcconnection.JdbcConnection;

public class AddBookLocation {

static Connection connection=null;

static CallableStatement cstmt=null;

public static void addBookLocation() throws SQLException

{

int fno,cno,rno,bno;

try

{

connection= JdbcConnection.getConnection();

cstmt = connection.prepareCall("call library.addBookLocations(?,?,?,?)");

Scanner sc=new Scanner(System.in);

System.out.println("Enter the floor number");

fno = sc.nextInt();

System.out.println("Enter the cupboard number");

cno = sc.nextInt();

System.out.println("Enter the rack number");

rno= sc.nextInt();

System.out.println("Enter the book id");

bno= sc.nextInt();

cstmt.setInt(1,fno);

cstmt.setInt(2,cno);

cstmt.setInt(3,rno);

cstmt.setInt(4,bno);

cstmt.executeUpdate();

System.out.println("You have updated Book Location");

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

connection.close();

cstmt.close();

}

}

}

**AddNewMember.java**

package library;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.Scanner;

import jdbcconnection.JdbcConnection;

public class AddNewMember

{

static Connection connection = null;

static CallableStatement callablestatement = null;

public static void addMember() throws SQLException

{

String first\_name;

String last\_name;

String pass;

int mobile;

String address;

Scanner scanner = new Scanner(System.in);

try

{

connection= JdbcConnection.getConnection();

callablestatement = connection.prepareCall("call library.addNewMember(?,?,?,?,?)");

System.out.println("Enter the first\_name : ");

first\_name =scanner.next();

System.out.println("Enter the last name : ");

last\_name = scanner.next();

System.out.println("Enter the password : ");

pass = scanner.next();

System.out.println("Enter the mobile number : ");

mobile = scanner.nextInt();

System.out.println("Enter the address : ");

address = scanner.next();

callablestatement.setString(1, first\_name);

callablestatement.setString(2, last\_name);

callablestatement.setString(3, pass);

callablestatement.setInt(4, mobile);

callablestatement.setString(5, address);

System.out.println(" New Member is added ");

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

connection.close();

callablestatement.close();

scanner.close();

}

}

}

**BookIssue.java**

package library;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Scanner;

import jdbcconnection.JdbcConnection;

public class BookIssue {

static Connection connection = null;

static CallableStatement callablestatement = null;

static ResultSet resultset = null;

public static void bookIssue() throws NoBookAvailableException,

SQLException, MaximumBookIssuedException {

Scanner scanner = new Scanner(System.in);

try {

int memid;

int bookid;

int result = 0;

connection = JdbcConnection.getConnection();

Statement statement = connection.createStatement();

System.out.println("Enter Member Id ");

memid = scanner.nextInt();

System.out.println("Enter Book Id ");

bookid = scanner.nextInt();

resultset = statement.executeQuery("select copies\_available from Books where book\_id="+ bookid);

while (resultset.next()) {

if (resultset.getInt(1) == 0)

throw new NoBookAvailableException(resultset.getInt(1));

}

resultset = statement

.executeQuery("select book\_issued from members where member\_id ="

+ memid);

while (resultset.next()) {

if (resultset.getInt(1) == 5) {

throw new MaximumBookIssuedException();

}

}

callablestatement = connection

.prepareCall("CALL library.bookIssues(?,?)");

callablestatement.setInt(1, memid);

callablestatement.setInt(2, bookid);

result = callablestatement.executeUpdate();

if (result == 0)

System.out.println("Book Issued!");

else

System.out.println("Can`t Issue!");

} catch (Exception e) {

System.out.println(e);

} finally {

connection.close();

scanner.close();

}

}

}

**DeleteBookLocation.java**

package library;

import java.sql.ResultSet;

import java.sql.CallableStatement;

import java.util.Scanner;

import java.sql.Connection;

import oracle.jdbc.OracleTypes;

import jdbcconnection.JdbcConnection;

public class DeleteBookLocations {

public static void DeleteBookLocations() {

CallableStatement callablestatement = null;

ResultSet resultset = null;

Scanner scanner = new Scanner(System.in);

Connection connection = null;

int book\_id, floor\_no, cupboard\_no, rack\_no;

try {

connection = JdbcConnection.getConnection();

callablestatement = connection

.prepareCall("call library.deleteBookLocations(?,?,?,?)");

System.out.println("Enter the floor number : ");

floor\_no = scanner.nextInt();

System.out.println("Enter the cupboard number : ");

cupboard\_no = scanner.nextInt();

System.out.println("Enter the rack number : ");

rack\_no = scanner.nextInt();

System.out.println("Enter the book id : ");

book\_id = scanner.nextInt();

callablestatement.setInt(1, floor\_no);

callablestatement.setInt(2, cupboard\_no);

callablestatement.setInt(3, rack\_no);

callablestatement.setInt(4, book\_id);

int value = callablestatement.executeUpdate();

if (value == 0) {

System.out.println(" record is deleted ");

} else

System.out.println("no record is deleted");

} catch (Exception e) {

System.out.println(e);

} finally {

try {

if (callablestatement != null)

callablestatement.close();

if (connection != null)

connection.close();

if (resultset != null)

resultset.close();

} catch (Exception e) {

System.out.println(e);

}

}

}

}

**MaximumBookIssuedException.java**

**package** library;

**public** **class** MaximumBookIssuedException **extends** Exception {

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

MaximumBookIssuedException() {

System.***out***.println("Maximum Book issued by member");

}

}

**NoBookAvailableException.java**

**package** library;

**public** **class** NoBookAvailableException **extends** Exception {

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

/\*\*

\*

\*/

NoBookAvailableException(**int** n) {

System.***out***

.println("All copies are already issued, copy is available in Library");

}

}

**ReceiptGeneration.java**

package library;

import java.sql.ResultSet;

import java.sql.Statement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.PreparedStatement;

public class ReceiptGeneration {

public static void ReceiptGenerate(int issueid) throws ClassNotFoundException, SQLException

{

try{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection connection = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","hr","hr");

//Statement statement = connection.createStatement();

PreparedStatement preparedstatement = connection.prepareStatement("select receipt\_id,issue\_date,return\_date,book\_name,first\_name,last\_name,fine from book\_issue bi,receipt r, books b, members m where r.member\_id = m.member\_id and r.book\_id = b.book\_id and r.issue\_id = bi.issue\_id and r.issue\_id = ?");

preparedstatement.setInt(1,issueid);

ResultSet resultset = preparedstatement.executeQuery();

while(resultset.next()){

System.out.println("Receipt for Late Return:");

System.out.println("");

System.out.println("");

System.out.print("Receipt No : ");

System.out.println(resultset.getInt(1));

System.out.println("Name : "+resultset.getString(5)+" "+resultset.getString(6));

System.out.println("Book Name\t\tIssue Date\t\tReturn Date\t\tAmount");

System.out.println("----------------------------------------------------------------------------------------------------");

System.out.println("");

System.out.println(resultset.getString(4)+"\t\t\t"+resultset.getDate(2)+"\t\t\t"+resultset.getDate(3)+"\t\t\t"+resultset.getInt(7));

}

}

catch(Exception e)

{

System.out.println(e);

}

finally{

//connection.close();

}

}

}

**ReturnBooks.java**

package library;

import java.sql.Date;

import java.sql.Statement;

import java.sql.CallableStatement;

import java.sql.Connection;

import jdbcconnection.JdbcConnection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Scanner;

import java.sql.PreparedStatement;

public class ReturnBooks extends ReceiptGeneration{

static Connection connection=null;

static Statement statement=null;

static CallableStatement callablestatement=null;

static ResultSet resultset=null;

public static void returnBook() throws SQLException {

int bookid,memid,issueid = 0;

Date expreturn=null,returndate = null;

Scanner scanner=new Scanner(System.in);

try

{

connection= JdbcConnection.getConnection();

Statement statement=connection.createStatement();

callablestatement=connection.prepareCall("call library.returnBooks(?, ?)");

System.out.println("Enter Member Id : ");

memid=scanner.nextInt();

System.out.println(" Enter Book Id : ");

bookid=scanner.nextInt();

PreparedStatement psmt = connection.prepareStatement("select issue\_id from book\_issue where member\_id=? and book\_id=? and return\_date is null");

psmt.setInt(1, memid);

psmt.setInt(2, bookid);

System.out.println(psmt.executeUpdate());

resultset = statement.executeQuery("select issue\_id from book\_issue where member\_id="+memid+"and book\_id="+bookid+"and return\_date is null");

while(resultset.next()){

issueid = resultset.getInt(1);

}

callablestatement.setInt(1, memid);

callablestatement.setInt(2,bookid);

callablestatement.executeUpdate();

resultset = statement.executeQuery("select exp\_return , return\_date from book\_issue where issue\_id ="+issueid);

while(resultset.next()){

returndate = resultset.getDate(2);

expreturn = resultset.getDate(1);

}

if(returndate.after(expreturn)){

ReceiptGenerate(issueid);

}

}

catch(Exception e)

{

System.out.print(e);

}

finally

{

connection.close();

}

}

}

**SearchBookLocation.java**

package library;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.SQLException;

import java.util.Scanner;

import oracle.jdbc.OracleTypes;

import jdbcconnection.JdbcConnection;

public class SearchBooklocation {

static CallableStatement callablestatement = null;

static Connection connection = null;

static ResultSet resultset = null;

public static void searchBooklocation()

{

Scanner scanner = new Scanner(System.in);

String book\_name;

try

{

connection= JdbcConnection.getConnection();

callablestatement = connection.prepareCall("call library.searchBooklocation(?,?)");

System.out.println("Enter the book name : ");

book\_name = scanner.next();

callablestatement.setString(1, book\_name);

callablestatement.registerOutParameter(2, OracleTypes.CURSOR);

callablestatement.executeUpdate();

resultset = (ResultSet) callablestatement.getObject(2);

System.out.println("BOOK ID\t\t\t"+"BOOK NAME \t\t\t"+"AUTHOR\t\t\t"+"PUBLICATION\t\t\t"+"FLOOR NUMBER\t\t"+"CUPBOARD NUMBER\t\t"+"RACK NUMBER");

while(resultset.next())

{

System.out.println(resultset.getString("book\_id")+"\t\t"+resultset.getString("book\_name")+"\t\t"+resultset.getString("author")

+"\t\t"+resultset.getString("publication")+"\t\t"+resultset.getInt("floor\_no")+"\t\t"+resultset.getInt("cupboard\_no")+"\t\t"+resultset.getInt("rack\_no"));

}

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

try{

if(callablestatement!=null)

callablestatement.close();

if(connection!=null)

connection.close();

if(resultset!=null)

resultset.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

}

}

**SearchByBookId.java**

package library;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.util.Scanner;

import oracle.jdbc.OracleTypes;

import jdbcconnection.JdbcConnection;

public class SearchBybookId {

static CallableStatement callablestatement = null;

static Connection connection = null;

static ResultSet resultset = null;

public static void searchBybookId() {

Scanner scanner = new Scanner(System.in);

try

{

int book\_id;

connection= JdbcConnection.getConnection();

callablestatement = connection.prepareCall("call library.searchBybookId(?,?)");

System.out.println("Enter the book id : ");

book\_id = scanner.nextInt();

callablestatement.setInt(1, book\_id);

callablestatement.registerOutParameter(2, OracleTypes.CURSOR);

callablestatement.executeUpdate();

resultset = (ResultSet) callablestatement.getObject(2);

System.out.println("BOOK NAME \t\t\t"+"AUTHOR\t\t\t"+"PUBLICATION\t\t\t"+"FLOOR NUMBER\t\t"+"CUPBOARD NUMBER\t\t"+"RACK NUMBER");

while(resultset.next())

{

System.out.println(resultset.getString("book\_name")+"\t\t"+resultset.getString("author")

+"\t\t"+resultset.getString("publication")+"\t\t"+resultset.getInt("floor\_no")+"\t\t"+resultset.getInt("cupboard\_no")+"\t\t"+resultset.getInt("rack\_no"));

}

}

catch(Exception e)

{

System.out.println(e);

}

finally{

try{

if(callablestatement!=null)

callablestatement.close();

if(connection!=null)

connection.close();

if(resultset!=null)

resultset.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

}

}

**SearchByMemberById.java**

package library;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.util.Scanner;

import oracle.jdbc.OracleTypes;

import jdbcconnection.JdbcConnection;

public class SearchMemberById

{

static CallableStatement callablestatement = null;

static Connection connection = null;

static ResultSet resultset = null;

public static void searchMemberById()

{

Scanner scanner = new Scanner(System.in);

int member\_id;

try

{

connection= JdbcConnection.getConnection();

callablestatement = connection.prepareCall("call library.searchMemberById(?,?)");

System.out.println("Enter the member id : ");

member\_id = scanner.nextInt();

callablestatement.setInt(1,member\_id);

callablestatement.registerOutParameter(2, OracleTypes.CURSOR);

callablestatement.executeUpdate();

resultset = (ResultSet) callablestatement.getObject(2);

System.out.println("MEMBER ID \t\t FIRST NAME \t\tLAST NAME \t\t MOBILE \t\tBOOK ISSUED ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

while(resultset.next())

{

System.out.println(resultset.getInt("member\_id")+"\t\t\t"+resultset.getString("first\_name")+"\t\t\t"

+resultset.getString("last\_name")+"\t\t\t"+resultset.getLong("mobile")+"\t\t\t"+resultset.getInt("book\_issued"));

}

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

try{

if(callablestatement!=null)

callablestatement.close();

if(connection!=null)

connection.close();

if(resultset!=null)

resultset.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

}

}

**SearchMemberByName.java**

package library;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Scanner;

import oracle.jdbc.OracleTypes;

import jdbcconnection.JdbcConnection;

public class SearchMemberByName {

static CallableStatement callablestatement = null;

static Connection connection = null;

static ResultSet resultset = null;

public static void SearchMemberByName() throws SQLException

{

Scanner scanner = new Scanner(System.in);

String membername;

try

{

connection= JdbcConnection.getConnection();

callablestatement = connection.prepareCall("call library.searchMemberByName(?,?)");

System.out.println("Enter the member first name OR last name : ");

membername = scanner.next();

callablestatement.setString(1,membername);

callablestatement.registerOutParameter(2, OracleTypes.CURSOR);

callablestatement.executeUpdate();

resultset = (ResultSet) callablestatement.getObject(2);

System.out.println("MEMBER ID \t\t FIRST NAME \t\tLAST NAME \t\t MOBILE \t\tBOOK ISSUED ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

while(resultset.next())

{

System.out.println(resultset.getInt("member\_id")+"\t\t\t"+resultset.getString("first\_name")+"\t\t\t"

+resultset.getString("last\_name")+"\t\t\t"+resultset.getLong("mobile")+"\t\t\t"+resultset.getInt("book\_issued"));

}

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

try{

if(callablestatement!= null)

callablestatement.close();

if(resultset!= null)

resultset.close();

if(connection!=null)

connection.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

}

}

**ViewIssuedBookRecord.java**

package library;

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.CallableStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Scanner;

import oracle.jdbc.OracleTypes;

import jdbcconnection.JdbcConnection;

public class ViewIssueBookRecord {

static CallableStatement callablestatement = null;

static Connection connection = null;

static ResultSet resultset = null;

public static void viewIssueBookRecord() throws SQLException

{

Scanner scanner = new Scanner(System.in);

int book\_id;

try

{

connection= JdbcConnection.getConnection();

callablestatement = connection.prepareCall("call library.viewIssueBookRecord(?,?)");

System.out.println("Enter the book id : ");

book\_id = scanner.nextInt();

callablestatement.setInt(1, book\_id);

callablestatement.registerOutParameter(2,OracleTypes.CURSOR);

callablestatement.executeUpdate();

resultset=(ResultSet) callablestatement.getObject(2);

System.out.println("BOOK NAME \t\t\t\t"+ "MEMBER ID\t\t\t"+"FIRST NAME\t\t\t"+"LAST NAME\t\t\t"+"EXP RETURN ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

while(resultset.next())

{

System.out.println(resultset.getString("book\_name")+"\t\t\t"+resultset.getInt("member\_id")+"\t\t\t"+

resultset.getString("first\_name")+"\t\t\t"+resultset.getString("last\_name")+"\t\t\t"+resultset.getDate("exp\_return"));

}

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

try{

if(callablestatement!=null)

callablestatement.close();

if(connection!=null)

connection.close();

if(resultset!=null)

resultset.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

}

}

**Members Package-**

**ViewIssuedBook.java**

**package** members;

**import** java.sql.CallableStatement;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Types;

**import** java.sql.ResultSet;

**import** java.util.Scanner;

**import** jdbcconnection.JdbcConnection;

**import** oracle.jdbc.OracleTypes;

**public** **class** ViewIssuedBook {

**static** Connection *connection* = **null**;

**static** CallableStatement *callablestatement* = **null**;

**static** ResultSet *resultset*;

**public** **static** **void** viewIssuedBook(**int** member\_id) **throws** ClassNotFoundException, SQLException {

**try**{

*connection* = JdbcConnection.*getConnection*();

*callablestatement* = *connection*.prepareCall("{CALL member.viewIssuedBook(?,?)}");

Scanner scanner = **new** Scanner(System.***in***);

*callablestatement*.setInt(1, member\_id);

*callablestatement*.registerOutParameter(2,OracleTypes.***CURSOR***);

*callablestatement*.executeUpdate();

*resultset*=(ResultSet) *callablestatement*.getObject(2);

System.***out***.println("BOOK NAME \t\t"+ "ISSUE DATE\t\t\t"+"EXP RETURN\t\t");

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

**while**(*resultset*.next())

{

System.***out***.println(*resultset*.getString("book\_name")+"\t\t\t"+*resultset*.getDate("issue\_date")+"\t\t\t"+*resultset*.getDate("exp\_return"));

System.***out***.println("");

}

}

**catch**(Exception e)

{

System.***out***.println(e);

}

**finally**

{

*callablestatement*.close();

*connection*.close();

}

}

}

**ViewHistory.java**

package members;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Types;

import java.util.Scanner;

import jdbcconnection.JdbcConnection;

import oracle.jdbc.OracleTypes;

public class ViewHistory

{

static Connection connection = null;

static CallableStatement callablestatement = null;

static ResultSet resultset;

public static void viewHistory(int member\_id) throws ClassNotFoundException, SQLException {

try

{

connection = JdbcConnection.getConnection();

callablestatement = connection.prepareCall("{CALL member.viewHistory(?,?)}");

Scanner scanner = new Scanner(System.in);

callablestatement.setInt(1, member\_id);

callablestatement.registerOutParameter(2,OracleTypes.CURSOR);

callablestatement.executeUpdate();

resultset=(ResultSet) callablestatement.getObject(2);

System.out.println("BOOK NAME \t\t\t\t\t\t"+ "ISSUE DATE\t\t\t\t\t"+"RETURN DATE\t\t");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

while(resultset.next())

{

System.out.println(resultset.getString("book\_name")+"\t\t\t\t\t"+resultset.getDate("issue\_date")+"\t\t\t\t\t"+resultset.getDate("return\_date"));

}

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

callablestatement.close();

connection.close();

}

}

}

**ViewPopularBook.java**

package members;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import jdbcconnection.JdbcConnection;

import oracle.jdbc.OracleTypes;

public class ViewPopularBook

{

static CallableStatement callablestatement = null;

static Connection connection = null;

static ResultSet resultset = null;

public static void viewPopularBook() throws ClassNotFoundException, SQLException

{

try

{

connection = JdbcConnection.getConnection();

callablestatement = connection.prepareCall("call member.viewPopularBook(?)");

callablestatement.registerOutParameter(1, OracleTypes.CURSOR);

callablestatement.executeUpdate();

resultset = (ResultSet) callablestatement.getObject(1);

System.out.println("BOOK NAME "+ " \t\t"+"AUTHOR " + "\t\t\t"+"publication");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

while(resultset.next())

{

System.out.println(resultset.getString("Book\_name")+"\t\t\t"+resultset.getString("author")+"\t\t\t"+resultset.getString("publication"));

System.out.println("");

}

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

try

{

if(callablestatement!=null)

callablestatement.close();

if(connection!= null)

connection.close();

if(resultset!=null)

resultset.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

}

}

**ChangePassword.java**

package members;

import java.sql.Connection;

import java.sql.CallableStatement;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.Scanner;

import jdbcconnection.JdbcConnection;

public class ChangePassword {

static Connection connection = null;

static CallableStatement callablestatement = null;

public static void changePassword(int member\_id) throws SQLException {

try {

String password1;

String confirmpassword;

connection = JdbcConnection.getConnection();

callablestatement = connection

.prepareCall("{CALL member.changePassword(?,?,?)}");

System.out.println("Enter password : ");

Scanner sc = new Scanner(System.in);

password1 = sc.next();

System.out.println("Enter password again : ");

confirmpassword = sc.next();

if (password1.equals(confirmpassword)) {

callablestatement.setString(1, password1);

callablestatement.setString(2, confirmpassword);

callablestatement.setInt(3, member\_id);

callablestatement.executeUpdate();

System.out.println("Password changed.");

} else {

System.out.println("Passwords do not match.");

}

} catch (Exception e) {

System.out.println(e);

} finally {

callablestatement.close();

connection.close();

}

}

}

**Driver Package-**

**LibrarianValidation.java**

package driver;

import jdbcconnection.JdbcConnection;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.util.Scanner;

public class LibrarianValidation {

static Connection connection = null;

static PreparedStatement pt = null;

public static int librarianValidate() throws SQLException {

int res = 0;

String username;

String pass;

Scanner scanner = new Scanner(System.in);

System.out.println("Enter Username : ");

username = scanner.next();

System.out.println("Enter Password : ");

pass = scanner.next();

connection = JdbcConnection.getConnection();

pt = connection.prepareStatement("select user\_name,password from librarian where user\_name=?");

pt.setString(1, username);

ResultSet rs = pt.executeQuery();

String orgUname = "", orPass = "";

while (rs.next()) {

orgUname = rs.getString("user\_name");

orPass = rs.getString("password");

} //end while

if (orPass.equals(pass))

{

System.out.println("login Successful :)");

rs.close();

return 1;

}

else{

System.out.println("Login Unsuccessful! Please try again! :(");

return -1;

}

}

}

**MemberValidation.java**

package driver;

import jdbcconnection.JdbcConnection;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Scanner;

public class MemberValidation {

static Connection connection = null;

static PreparedStatement preparedstatement = null;

protected int validatemem() throws SQLException {

int memberid=0;

String pass="";

Scanner scanner = new Scanner(System.in);

System.out.println("Enter Member Id : ");

memberid = scanner.nextInt();

System.out.println("Enter Password : ");

pass = scanner.next();

connection = JdbcConnection.getConnection();

preparedstatement = connection.prepareStatement("select member\_id,password from members where member\_id=?");

preparedstatement.setInt(1,memberid);

ResultSet rs = preparedstatement.executeQuery();

int memberId = 0;

String orPass = "";

while (rs.next()) {

memberId = rs.getInt("member\_id");

orPass = rs.getString("password");

} //end while

if (orPass.equals(pass))

{

System.out.println("login Successful :)");

rs.close();

return memberid;

}

else{

System.out.println("Login Unsuccessful! Please try again! :(");

return -1;

}

}

}

**LibraryManagementSystem.java**

package driver;

import java.sql.SQLException;

import java.util.Scanner;

import library.AddBook;

import library.AddBookLocation;

import library.AddNewMember;

import library.BookIssue;

import library.DeleteBookLocations;

import library.MaximumBookIssuedException;

import library.NoBookAvailableException;

import library.ReturnBooks;

import library.SearchBooklocation;

import library.SearchBybookId;

import library.SearchMemberById;

import library.SearchMemberByName;

import library.ViewIssueBookRecord;

import members.ChangePassword;

import members.ViewHistory;

import members.ViewIssuedBook;

import members.ViewPopularBook;

//import jdbcconnection.JdbcConnection;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class LibraryManagementSystem

{

public static void main(String[] args) throws SQLException, NoBookAvailableException, MaximumBookIssuedException, ClassNotFoundException, IOException

{

Scanner scanner = new Scanner(System.in);

BufferedReader bufferedreader = new BufferedReader(new InputStreamReader(System.in));

int choice\_as\_lib\_Member, choice\_as\_lib=0,choice\_as\_member,member\_id,libvalue;

int search\_book,search\_member;

do{

System.out.println("1.MEMBER \t\t\t 2.LIBRARIAN\t\t\t 3.EXIT ");

choice\_as\_lib\_Member = scanner.nextInt();

switch(choice\_as\_lib\_Member)

{

case 1 : MemberValidation membervalidation = new MemberValidation();

member\_id = membervalidation.validatemem();

if(member\_id != -1)

{

do

{

System.out.println("WELCOME TO MEMBER INTERFACE ");

System.out.println("1.View History\n2.View Issued Book\n3.View popular book\n4.Change password\n5.EXIT");

choice\_as\_member = scanner.nextInt();

switch(choice\_as\_member)

{

case 1 :ViewHistory.viewHistory(member\_id);

break;

case 2:ViewIssuedBook.viewIssuedBook(member\_id);

break;

case 3:ViewPopularBook.viewPopularBook();

break;

case 4:ChangePassword.changePassword(member\_id);

break;

case 5 :System.exit(0);

}

}while(choice\_as\_member!=5);

}

break;

case 2 :libvalue = LibrarianValidation.librarianValidate();

if(libvalue == 1){

do

{

System.out.println();

System.out.println("WELCOME TO LIBRARIAN INTERFACE ");

System.out.println("1.Book Issue\n2.Search book\n3.Delete book location\n4.Search Member\n5.View issued book\n6.Add Book\n7.Add Book Location\n8.Return Book\n9.Add New Member\n10.Exit");

choice\_as\_lib = scanner.nextInt();

switch(choice\_as\_lib)

{

case 1 :BookIssue.bookIssue();

break;

case 2 :System.out.println();

System.out.println("1. Search by Name \t\t 2.Search by ID \t\t 3.EXIT");

search\_book = scanner.nextInt();

switch(search\_book)

{

case 1 : SearchBooklocation.searchBooklocation();

break;

case 2 : SearchBybookId.searchBybookId();

break;

case 3 :break;

}

case 3 :DeleteBookLocations.DeleteBookLocations();

break ;

case 4 :System.out.println();

System.out.println("1. Search by Name \t\t 2.Search by ID \t\t 3.EXIT");

search\_member = scanner.nextInt();

switch(search\_member)

{

case 1 : SearchMemberByName.SearchMemberByName();

break;

case 2 : SearchMemberById.searchMemberById();

break;

case 3 :break;

}

case 5 :ViewIssueBookRecord.viewIssueBookRecord();

break;

case 6:AddBook.addBook();

break;

case 7: AddBookLocation.addBookLocation();

break;

case 8: ReturnBooks.returnBook();

break;

case 9: AddNewMember.addMember();

case 10:System.exit(0);

}

}while(choice\_as\_lib!=10);

}

case 3 :System.exit(0);

}

}while(choice\_as\_lib\_Member!=3);

}

}