

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help
JOBBC src com company Main main
Main.java x test.java x Allocate.java x Use.java x Drive.java x Book.java x
package com.company;
import java.sql.SQLException;
import java.util.Scanner;

public class Main {

    public static void call()
    {
        System.out.println("Enter the key");
        System.out.println(" 0 -- exit\n 1 -- User Details \n 2 -- Book Details\n 3 -- Allocate Book\n 4 -- Book Return" );
    }

    public static void main(String[] args) throws SQLException {
        Scanner scan = new Scanner(System.in);
        System.out.println(" 1 -- User Details \n 2 -- Book Details \n 3 -- Allocate Book\n 4 -- Book Return");

        boolean main_flag = true;
        while (main_flag) {
            int main_key = scan.nextInt();
            switch (main_key) {
                case 0 -> main_flag = false;

                case 1 -> {
                    user.userDetails();
                    call();
                    break;
                }
            }
        }
    }
}
```

FileEditViewNavigateCodeRefactorBuildRunToolsVCSWindowDB NavigatorHelpAllocate.java - Main.java

JDBCsrccomcompanyMainmain

ProjectMain.javatest.javaAllocate.javaUser.javaDriver.javaBook.java

DB Browser

Structure

Bookmarks

```
31 }
32
33 case (2) -> {
34     Book.bookDetails();
35     call();
36     break;
37 }
38
39 case (3) -> {
40     System.out.println("Enter the Mobile No.");
41     String mobile= scan.next();
42     if (mobile.length()==10)
43     {
44         if (!User.CheckValidUser(mobile)) {
45             System.out.println(" Mobile No not register \n Please Register");
46             User.UserInsert();
47         }
48
49         if (Allocate.checkBookAvailable(mobile)) {
50             System.out.println("Enter the Book Name");
51             String bookName = scan.next();
52             boolean increment = true;
53             Allocate.insertUserBook(mobile, bookName);
54             Allocate.changeBookCount(mobile, increment);
55             Book.changeBookCount(bookName, !increment);
56             } else System.out.println("you reached the Max Limit!");
57         }
58     }
59
60 else
61     System.out.println("Enter a Valid Number");
62     call();
63     break;
64 }
65
66 case(4) ->{
```



Main

File Edit View Navigate Code Reflector Build Run Tools VCS Window DB Navigator Help Allocate.java Main.java

JDGC src com company Main Allocate.java User.java Driver.java Book.java

Main.java test.java Allocate.java User.java Driver.java Book.java

```
58 System.out.println("Enter a Valid Number");
59 call();
60 break;
61 }
62
63 case(4) ->{
64     System.out.println("Enter the Mobile No");
65     String mobile = scan.next();
66     if (User.CheckValidUser(mobile)){
67         System.out.println("Enter the Book Want to return");
68         String bookName = scan.next();
69         if(Allocate.checkBookPresence(mobile,bookName)){
70             boolean increment = false;
71             Allocate.checkForFine(mobile,bookName);
72             Book.changeBookCount(bookName,increment);
73             Allocate.deleteUserBook(mobile,bookName);
74             Allocate.changeBookCount(mobile,increment);
75         }
76     }
77     else System.out.println("Enter the Correct Book Name!!");
78 }
79 else{
80     System.out.println("Enter a Valid Number");}
81 call();
82 break;
83 }
84 default ->{
85     System.out.println("Enter a Valid Key");
86     call();
87 }
88 }
89 }
90 }
```

```
package com.company;
```

```
import java.time.LocalDate;
```

```
import java.sql.*;
```

```
import static java.time.temporal.ChronoUnit.DAYS;
```

```
public class Allocate {
```

```
    private static Connection connection() throws SQLException {
```

```
        String url = "jdbc:mysql://localhost:3306/user";
```

```
        String username = "root";
```

```
        String password = "HgciB12025GRobotic";
```

```
        return DriverManager.getConnection(url, username, password);
```

```
    static boolean checkBookAvailable(String mobile) throws SQLException {
```

```
        int maxBookAvailable = 5;
```

```
        Connection con = connection();
```

```
        String query = "SELECT BookSingingCount FROM user_details WHERE Mobile = ?";
```

```
        PreparedStatement statement1 = con.prepareStatement(query);
```

```
        statement1.setString(1, mobile);
```

```
        ResultSet resultSet = statement1.executeQuery();
```

```
        resultSet.next();
```

```
        int bookAvailable = resultSet.getInt("columnLabel: 'BookSingingCount'");
```

```
        con.close();
```

```
        return bookAvailable < maxBookAvailable;
```

```
}
```

```
static void changeBookCount(String mobile, boolean increment) throws SQLException {
```

```
    Connection con = connection();
```

```
    String query = "select * from user_details Where Mobile = ?";
```



```

31 static void changeBookCount(String mobile, boolean increment) throws SQLException {
32     Connection con = connection();
33     String query = "select * from user_details where Mobile = ?";
34     PreparedStatement statement1 = con.prepareStatement(query);
35     statement1.setString(1, mobile);
36     ResultSet resultSet = statement1.executeQuery();
37     resultSet.next();
38
39     int curCount;
40     if (increment) {
41         curCount = resultSet.getInt(columnLabel "BookSingingCount") + 1;
42     } else {
43         curCount = resultSet.getInt(columnLabel "BookSingingCount") - 1;
44     }
45
46     String query1 = "update user_details set BookSingingCount = ? where Mobile = ?";
47     PreparedStatement statement2 = con.prepareStatement(query1);
48     statement2.setInt(1, curCount);
49     statement2.setString(2, mobile);
50     statement2.executeUpdate();
51     con.close();
52     System.out.println("User Book Details Updated Successfully");
53 }
54
55 static void insertUserBook(String mobile, String bookName) throws SQLException {
56     Connection con = connection();
57     String query = ("INSERT INTO bookmanager(Mobile,BookName,ReturnDate) values(?,?,?)");
58     LocalDate retunDate = LocalDate.now().plusDays(10);
59     PreparedStatement statement1 = con.prepareStatement(query);
60     statement1.setString(1, mobile);
61     statement1.setDate(2, Date.valueOf(retunDate));
62     statement1.setString(3, bookName);
63     statement1.executeUpdate();

```



```

55 static void insertUserBook(String mobile, String bookName) throws SQLException {
56     Connection con = connection();
57     String query = ("INSERT INTO bookmanager(Mobile,BookName,ReturnDate) values(?, ?, ?)");
58     LocalDateTime returnDate = LocalDateTime.now().plusDays(10);
59     PreparedStatement statement1 = con.prepareStatement(query);
60     statement1.setString(1, mobile);
61     statement1.setDate(2, Date.valueOf(returnDate));
62     statement1.setString(3, bookName);
63     statement1.executeUpdate();
64     con.close();
65 }
66
67 static boolean checkBookPresence(String mobile, String bookName) throws SQLException {
68     Connection con = connection();
69     String query = ("SELECT * FROM bookmanager WHERE Mobile = ? AND BookName = ?");
70     PreparedStatement statement1 = con.prepareStatement(query);
71     statement1.setString(1, mobile);
72     statement1.setString(2, bookName);
73     ResultSet resultSet = statement1.executeQuery();
74     boolean result = resultSet.next();
75     con.close();
76     return result;
77 }
78
79 static void deleteUserBook(String mobile, String bookName) throws SQLException {
80     Connection con = connection();
81     String query = ("DELETE FROM bookmanager WHERE Mobile = ? AND BookName = ?");
82     PreparedStatement statement1 = con.prepareStatement(query);
83     statement1.setString(1, mobile);
84     statement1.setString(2, bookName);
85     statement1.executeUpdate();
86     con.close();
87 }

```

```

75         con.close();
76         return result;
77     }
78
79     static void deleteUserBook(String mobile, String bookName) throws SQLException {
80         Connection con = connection();
81         String query = ("DELETE FROM bookmanager WHERE Mobile = ? AND bookName = ?");
82         PreparedStatement statement1 = con.prepareStatement(query);
83         statement1.setString(1, mobile);
84         statement1.setString(2, bookName);
85         statement1.executeUpdate();
86         con.close();
87     }
88
89     static void checkForFine(String mobile, String bookName) throws SQLException {
90         Connection con = connection();
91         String query = ("SELECT ReturnDate FROM bookmanager WHERE Mobile = ? AND bookName = ?");
92         PreparedStatement statement1 = con.prepareStatement(query);
93         statement1.setString(1, mobile);
94         statement1.setString(2, bookName);
95         ResultSet resultSet = statement1.executeQuery();
96         resultSet.next();
97         LocalDate date = LocalDate.parse(resultSet.getString(1, "ReturnDate"));
98         LocalDate today = LocalDate.now();
99         long dayDifference = date.until(today, DAYS);
100         if (dayDifference > 0) System.out.println("You surpassed Date Limit!\nFine Amount : "+dayDifference*4);
101     }
102 }

```




```

1  package com.company;
2  import java.sql.*;
3  import java.util.Scanner;
4
5  public class User {
6      static Scanner scan = new Scanner(System.in);
7
8      static void call()
9      {
10         System.out.println("Enter the key to perform operation in user details");
11         System.out.println("0 -- exit\n 1 -- View User data\n 2 -- Insert User data\n 3 -- Delete User data\n 4 -- All Available User\n 5 -- Check Valid User");
12     }
13
14     private static Connection connection() throws SQLException {
15         String url = "jdbc:mysql://localhost:3306/user";
16         String username = "root";
17         String password = "Mgcib12025@Robotic";
18         return DriverManager.getConnection(url, username, password);
19     }
20
21     static void UserInsert() throws SQLException {
22         System.out.println("Enter the Name");
23         String name = scan.next();
24         System.out.println("Enter the mobile_No");
25         String mobile_no = scan.next();
26         Connection con = connection();
27
28         if (mobile_no.length() == 10) {
29             try {
30                 String query = "insert into user_details(Name, Mobile) values(?, ?)";
31                 PreparedStatement statement1 = con.prepareStatement(query);

```



```
static void UserInsert() throws SQLException {
    System.out.println("Enter the Name");
    String name = scan.next();
    System.out.println("Enter the mobile_no");
    String mobile_no = scan.next();
    Connection con = connection();
```

```
if (mobile_no.length() == 10) {
    try {
        String query = "insert into user_details(Name, Mobile) values(?, ?)";
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString(1, name);
        statement1.setString(2, mobile_no);
        statement1.executeUpdate();
        System.out.println("Added Successfully");
        ViewUser(mobile_no);
    }
    catch (Exception e) {
        System.out.println("Number Already exist");
    }
}
```

```
else {
    System.out.println("Enter a 10 digit Number");
}
con.close();
```

```
static void ViewUser(String mobile) {
    try {
        String query = "select * from user_details Where Mobile = ?";
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help AllocateJava - User.java
JDBC src \ com \ company \ User \ ViewAllUser
Main.java test.java Allocate.java User.java Driver.java Book.java
}
static void ViewUser(String mobile) {
    try {
        String query = "select * from user_details Where Mobile = ?";
        Connection con = connection();
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString(1, mobile);
        ResultSet resultSet = statement1.executeQuery();
        resultSet.next();
        String name = resultSet.getString(columnLabel: "Name");
        int user_id = resultSet.getInt(columnLabel: "UserId");
        int curHaveBookCount = resultSet.getInt(columnLabel: "BookUsingCount");
        System.out.printf("%d %s %s Current Book Holdings -- %d", user_id, name, mobile, curHaveBookCount).println();
        if (curHaveBookCount > 0) {
            String querybook = "SELECT * FROM bookmanager WHERE Mobile = ?";
            PreparedStatement statement2 = con.prepareStatement(querybook);
            statement2.setString(1, mobile);
            ResultSet bookresult = statement2.executeQuery();
            while (bookresult.next()) {
                String bookName = bookresult.getString(columnLabel: "BookName");
                String returnDate = bookresult.getString(columnLabel: "ReturnDate");
                System.out.printf("%s ReturnDate -- %s\n", bookName, returnDate);
            }
            System.out.println();
        }
        con.close();
    } catch (SQLException e) {
        System.out.println("Enter a Valid Number!");
    }
}
static void UserDelete() throws SQLException {
    Version Control Problems TOOO Terminal Build
Download one-built shared toolchain...
```


Main.java Allocate.java User.java Driver.java Book.java
 System.out.println("Enter a valid Number");

```

    }
}

static void UserDelete() throws SQLException {
    Connection con = connection();
    System.out.println("Enter the Mobile_No to delete");
    String mobile = scnn.next();
    if (checkValidUser(mobile)) {
        PreparedStatement statement1 = con.prepareStatement("delete from user_details where Mobile = ?");
        statement1.setString(1, mobile);
        statement1.executeUpdate();
        System.out.println("Deleted Successfully");
        con.close();
    }
    else System.out.println("Enter a Valid Number");
}

static void ViewAllUser() throws SQLException {
    Connection con = connection();
    PreparedStatement statement1 = con.prepareStatement("select * from user_details");
    ResultSet resultSet = statement1.executeQuery();

    while (resultSet.next()) {
        int id = resultSet.getInt("UserId");
        String name = resultSet.getString("Name");
        String mobile = resultSet.getString("Mobile");
        int curHaveBookCount = resultSet.getInt("BookUsingCount");
        System.out.printf("%d %s %s Current Book Holdings -- %d", id, name, mobile, curHaveBookCount).println();
        if (curHaveBookCount > 0) {
            String querybook = "select * from bookmanager WHERE Mobile = ?";
            PreparedStatement statement2 = con.prepareStatement(querybook);
            statement2.setString(1, mobile);
            ResultSet bookresult = statement2.executeQuery();
            while (bookresult.next()) {

```



```

96 static void ViewAllUser() throws SQLException{
97     Connection con = connection();
98     PreparedStatement statement = con.prepareStatement( sql "select * from user_details");
99     ResultSet resultSet = statement.executeQuery();
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128

```

```

        while (resultSet.next()) {
            I
            int id = resultSet.getInt( columnLabel: "UserId");
            String name = resultSet.getString( columnLabel: "Name");
            String mobile = resultSet.getString( columnLabel: "Mobile");
            int curHaveBookCount = resultSet.getInt( columnLabel: "BookUsingCount");
            System.out.printf("\n%d %s %s Current Book Holdings -- %d", id, name, mobile, curHaveBookCount).println();
            if (curHaveBookCount > 0) {
                String queryBook = "SELECT * FROM bookmanager WHERE Mobile = ?";
                PreparedStatement statement2 = con.prepareStatement(queryBook);
                statement2.setString( parameterIndex: 1, mobile);
                ResultSet bookresult = statement2.executeQuery();
                while (bookresult.next()) {
                    String bookName = bookresult.getString( columnLabel: "BookName");
                    String returnDate = bookresult.getString( columnLabel: "ReturnDate");
                    System.out.printf("%5s ReturnDate -- %s\n", bookName, returnDate);
                }
            }
            System.out.println();
            con.close();
        }
    }
}

```

```

    static boolean CheckValidUser(String mobile) throws SQLException{
        String query = "SELECT Mobile from user_details Where Mobile = ?";
        Connection con = connection();
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString( parameterIndex: 1,mobile);
        ResultSet resultSet = statement1.executeQuery();
        boolean result = resultSet.next();
    }
}

```



```

130 con.close();
131 }
132
133 static boolean CheckValidUser(String mobile) throws SQLException{
134     String query = "SELECT Mobile from user_details Where Mobile = ?";
135     Connection con = connection;
136     PreparedStatement statement1 = con.prepareStatement(query);
137     statement1.setString(1, mobile);
138     ResultSet resultSet = statement1.executeQuery();
139     boolean result = resultSet.next();
140     con.close();
141     return result;
142 }
143
144 static void userDetails() throws SQLException {
145     System.out.println("Entered the User Details part");
146     System.out.println("1 -- View User data\n 2 -- Insert User data\n 3 -- Delete User data\n 4 -- All Available User\n 5 -- Check Valid User");
147     boolean flag1 = true;
148
149     while (flag1) {
150         int key = scan.nextInt();
151         switch (key) {
152             case 0 -> flag1 = false;
153             case 1 -> {
154                 System.out.println("Enter the Mobile No to view the user");
155                 String mobile = scan.next();
156                 ViewUser(mobile);
157                 call1();
158             }
159             case 2 -> {
160                 UserInsert();
161                 call1();
162             }
163         }
164     }
165 }

```

```
call();  
break;  
}  
case 2 -> {  
    UserInsert();  
    call();  
    break;  
}  
case 3 -> {  
    UserDelete();  
    call();  
    break;  
}  
case 4 -> {  
    ViewAllUser();  
    call();  
    break;  
}  
case 5 -> {  
    System.out.println("Enter User Mobile No: ");  
    String mobile = scan.next();  
    if(CheckValidUser(mobile)) {  
        System.out.println("User Details...");  
        ViewUser(mobile);  
    }  
    else  
        System.out.println("Not a Valid Mobile No!");  
    call();  
}  
default -> {  
    System.out.println("Enter a Valid User Key");  
    call();  
}
```


File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help Allocate Java Book.java
JDBC src: com: company: Book connection
Main.java x test.java x Allocate.java x Use.java x DriverManager x Book.java
package com.company;

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

public class Book {
 static Scanner sc = new Scanner(System.in);
 public static void call2() {
 System.out.println("Enter the Key to perform operation in Book Details");
 System.out.println("0 -- exit\n1 -- View Book data\n2 -- Insert Book data\n3 -- To View all the available Book ");
 }
 private static Connection connection() throws SQLException {
 String url = "jdbc:mysql://localhost:3306/user";
 String username = "root";
 String password = "Mgc12825@Robotic";
 return DriverManager.getConnection(url, username, password);
 }
 static void ViewBook() throws SQLException {
 System.out.println("Enter the Book Name to view");
 String name = sc.next();
 String query = "select * from book_details Where Book_Name = ?";
 Connection con = connection();
 PreparedStatement statement = con.prepareStatement(query);
 statement.setString(1, name);
 ResultSet resultSet = statement.executeQuery();
 resultSet.next();
 int isbn = resultSet.getInt("columnlabel:ISBN");
 int curCount = resultSet.getInt("columnlabel:BookCount");
 int maxCount = resultSet.getInt("columnlabel:MaxCount");
 System.out.printf("ISBN = %d, Name = %s, CurCount = %d, MaxCount = %d, isbn, name, curCount, maxCount).println();
 }
}

```

19
20
21 static void ViewBook() throws SQLException {
22     System.out.println("Enter the Book Name to view");
23     String name = scan.next();
24     String query = "select * from book_details where Book_Name = ?";
25     Connection con = connection();
26     PreparedStatement statement1 = con.prepareStatement(query);
27     statement1.setString(1, name);
28     ResultSet resultSet = statement1.executeQuery();
29     resultSet.next();
30     int isbn = resultSet.getInt("ISBN");
31     int curCount = resultSet.getInt("BookCount");
32     int maxCount = resultSet.getInt("MaxCount");
33     System.out.printf("ISBN = %d, Name = %s, CurCount = %d, MaxCount = %d", isbn, name, curCount, maxCount);
34     con.close();
35 }
36
37 static void InsertBook() throws SQLException {
38     System.out.println("Enter the Name");
39     String name = scan.next();
40     System.out.println("Enter the Count of Books");
41     int count = scan.nextInt();
42     String query = "insert into book_details(Book_Name, BookCount, MaxCount) values(?, ?, ?)";
43     Connection con = connection();
44     PreparedStatement statement1 = con.prepareStatement(query);
45     statement1.setString(1, name);
46     statement1.setInt(2, count);
47     statement1.setInt(3, count);
48     statement1.executeUpdate();
49     System.out.println("Book Added to the Inventory Successfully.");
50     con.close();
51 }

```



```

static void InsertBook() throws SQLException{
    System.out.println("Enter the Name");
    String name = sccon.next();
    System.out.println("Enter the Count of Books");
    int count = sccon.nextInt();
    String query = "insert into book_details(Book_Name,BookCount,MaxCount) values(?, ?, ?)";
    Connection con = connection();
    PreparedStatement statement1 = con.prepareStatement(query);
    statement1.setString(1, name);
    statement1.setInt(2, count);
    statement1.setInt(3, count);
    statement1.executeUpdate();
    System.out.println("Book Added to the Inventory Successfully.");
    con.close();
}

static void ViewAllBooks() throws SQLException{
    Connection con = connection();
    PreparedStatement statement1 = con.prepareStatement("select * from book_details");
    ResultSet resultSet = statement1.executeQuery();
    System.out.println("ISBN\tNAME\tCURCOUNT\tMAXCOUNT\n");
    while (resultSet.next()) {
        String isbn = resultSet.getString(1);
        String name = resultSet.getString(2);
        int curCount = resultSet.getInt(3);
        int maxCount = resultSet.getInt(4);
        System.out.printf("%s\t%s\t%d\t%d\n", isbn, name, curCount, maxCount);
    }
}

```

```

    }

    static void changeBookCount(String bookName, boolean increment) throws SQLException{
        Connection con = connection();
        String query = "select * from book_details Where Book_Name = ?";
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString(1, bookName);
        ResultSet resultSet = statement1.executeQuery();

        resultSet.next();
        int curCount;
        if (increment) {
            curCount = resultSet.getInt(1);
            System.out.println("Book count Incremented Successfully");
        }
        else{
            curCount = resultSet.getInt(1);
            System.out.println("Book Count Decremented Successfully");
        }

        String query1 = "update book_details set BookCount = ? Where Book_Name = ?";
        PreparedStatement statement2 = con.prepareStatement(query1);
        statement2.setInt(1, curCount);
        statement2.setString(2, bookName);
        statement2.executeUpdate();
        con.close();
    }

    static void bookDetails() throws SQLException {
        System.out.println("Entered the Book Details part");
        System.out.println("1 -- View Book data\n 2 -- Insert Book data\n 3 -- To View all the available Book");
        boolean flag2 = true;
    }
}

```



```
94 static void bookDetails() throws SQLException {
95     System.out.println("Entered the Book Details part");
96     System.out.println(" 1 -- View Book data\n 2 -- Insert Book data\n 3 -- To View all the available Book ");
97     boolean flag2 = true;
98 }
```

```
99 while (flag2) {
```

```
100     int key = scan.nextInt();
```

```
101     switch (key) {
```

```
102         case 0 -> flag2 = false;
```

```
103         case 1 -> {
```

```
104             ViewBook();
```

```
105             call2();
```

```
106             break;
```

```
107         }
```

```
108         case 2 -> {
```

```
109             InsertBook();
```

```
110             call2();
```

```
111             break;
```

```
112         }
```

```
113         case 3 -> {
```

```
114             ViewAllBooks();
```

```
115             call2();
```

```
116             break;
```

```
117         }
```

```
118         default -> {
```

```
119             System.out.println("Enter a Valid Book Key ");
```

```
120             call2();
```

```
121         }
```

```
122     }
```

```
123 }
```

Version Control

Problems

TOC

Terminal

Build