

```
1 import java.sql.DriverManager;
2 import java.sql.SQLException;
3 import java.sql.Connection;
4 import java.util.Scanner;
5 import java.sql.Statement;
6 import java.sql.ResultSet;
7
8
9
10 class HotelReservationSystem {
11     private static final String url = "jdbc:mysql://
localhost:3306/hotel_db";
12     private static final String username = "root";
13     private static final String password = "7491";
14
15     public static void main(String[] args) throws
ClassNotFoundException, SQLException {
16         try{
17             Class.forName("com.mysql.cj.jdbc.Driver"
);
18         }catch (ClassNotFoundException e){
19             System.out.println(e.getMessage());
20         }
21
22         try{
23             Connection connection = DriverManager.
getConnection(url, username, password);
24             while(true){
25                 System.out.println();
26                 System.out.println("HOTEL MANAGEMENT
SYSTEM");
27                 Scanner scanner = new Scanner(System.
in);
28                 System.out.println("1. Reserve a room
");
29                 System.out.println("2. View
Reservations");
30                 System.out.println("3. Get Room
Number");
31                 System.out.println("4. Update
Reservations");
```

```

32         System.out.println("5. Delete
    Reservations");
33         System.out.println("0. Exit");
34         System.out.print("Choose an option: "
    );
35         int choice = scanner.nextInt();
36         switch (choice) {
37             case 1:
38                 reserveRoom(connection,
    scanner);
39                 break;
40             case 2:
41                 viewReservations(connection);
42                 break;
43             case 3:
44                 getRoomNumber(connection,
    scanner);
45                 break;
46             case 4:
47                 updateReservation(connection
    , scanner);
48                 break;
49             case 5:
50                 deleteReservation(connection
    , scanner);
51                 break;
52             case 0:
53                 exit();
54                 scanner.close();
55                 return;
56             default:
57                 System.out.println("Invalid
    choice. Try again.");
58         }
59     }
60
61     }catch (SQLException e){
62         System.out.println(e.getMessage());
63     } catch (InterruptedException e) {
64         throw new RuntimeException(e);
65     }

```

```

66
67
68     }
69
70     private static void reserveRoom(Connection
connection, Scanner scanner) {
71         try {
72             System.out.print("Enter guest name: ");
73             String guestName = scanner.next();
74             scanner.nextLine();
75             System.out.print("Enter room number: ");
76             int roomNumber = scanner.nextInt();
77             System.out.print("Enter contact number
: ");
78             String contactNumber = scanner.next();
79
80             String sql = "INSERT INTO reservation (
guest_name, room_number, contact_number) " +
81                 "VALUES ('" + guestName + "', "
+ roomNumber + ", '" + contactNumber + "')";
82
83             try (Statement statement = connection.
createStatement()) {
84                 int affectedRows = statement.
executeUpdate(sql);
85
86                 if (affectedRows > 0) {
87                     System.out.println("Reservation
successful!");
88                 } else {
89                     System.out.println("Reservation
failed.");
90                 }
91             }
92         } catch (SQLException e) {
93             e.printStackTrace();
94         }
95     }
96
97     private static void viewReservations(Connection
connection) throws SQLException {

```

```

98         String sql = "SELECT reservation_id,
    guest_name, room_number, contact_number,
    reservation_date FROM reservation";
99
100        try (Statement statement = connection.
    createStatement());
101            ResultSet resultSet = statement.
    executeQuery(sql)) {
102
103            System.out.println("Current Reservations
    :");
104            System.out.println(
    "+-----+-----+-----+-----+
    +-----+-----+-----+-----+
    ");
105            System.out.println("| Reservation ID |
    Guest      | Room Number  | Contact Number
    | Reservation Date      |");
106            System.out.println(
    "+-----+-----+-----+-----+
    +-----+-----+-----+-----+
    ");
107
108            while (resultSet.next()) {
109                int reservationId = resultSet.getInt(
    "reservation_id");
110                String guestName = resultSet.
    getString("guest_name");
111                int roomNumber = resultSet.getInt("
    room_number");
112                String contactNumber = resultSet.
    getString("contact_number");
113                String reservationDate = resultSet.
    getTimestamp("reservation_date").toString();
114
115                // Format and display the
    reservation data in a table-like format
116                System.out.printf("| %-14d | %-15s
    | %-13d | %-20s | %-19s   |\n",
117                reservationId, guestName,
    roomNumber, contactNumber, reservationDate);

```

```

118         }
119
120         System.out.println(
121             "+-----+-----+-----+-----+
122             +-----+-----+-----+-----+"
123         );
124     }
125
126     private static void getRoomNumber(Connection
127         connection, Scanner scanner) {
128         try {
129             System.out.print("Enter reservation ID
130             : ");
131             int reservationId = scanner.nextInt();
132             System.out.print("Enter guest name: ");
133             String guestName = scanner.next();
134
135             String sql = "SELECT room_number FROM
136             reservation " +
137             "WHERE reservation_id = " +
138             reservationId +
139             " AND guest_name = '" +
140             guestName + "'";
141
142             try (Statement statement = connection.
143                 createStatement();
144                 ResultSet resultSet = statement.
145                     executeQuery(sql)) {
146                 if (resultSet.next()) {
147                     int roomNumber = resultSet.
148                         getInt("room_number");
149                     System.out.println("Room number
150                     for Reservation ID " + reservationId +
151                     " and Guest " +
152                     guestName + " is: " + roomNumber);
153                 } else {
154                     System.out.println("Reservation
155                     not found for the given ID and guest name.");
156                 }
157             }
158         } catch (SQLException e) {
159             e.printStackTrace();
160         }
161     }
162 }

```

```

145         }
146     }
147     } catch (SQLException e) {
148         e.printStackTrace();
149     }
150 }
151 private static void updateReservation(Connection
connection, Scanner scanner) {
152     try {
153         System.out.print("Enter reservation ID
to update: ");
154         int reservationId = scanner.nextInt();
155         scanner.nextLine(); // Consume the
newline character
156
157         if (!reservationExists(connection,
reservationId)) {
158             System.out.println("Reservation not
found for the given ID.");
159             return;
160         }
161
162         System.out.print("Enter new guest name
: ");
163         String newGuestName = scanner.nextLine
();
164         System.out.print("Enter new room number
: ");
165         int newRoomNumber = scanner.nextInt();
166         System.out.print("Enter new contact
number: ");
167         String newContactNumber = scanner.next
();
168
169         String sql = "UPDATE reservation SET
guest_name = '" + newGuestName + "', " +
170             "room_number = " + newRoomNumber
+ ", " +
171             "contact_number = '" +
newContactNumber + "' " +
172             "WHERE reservation_id = " +

```

```
172 reservationId;
173
174         try (Statement statement = connection.
createStatement()) {
175             int affectedRows = statement.
executeUpdate(sql);
176
177             if (affectedRows > 0) {
178                 System.out.println("Reservation
updated successfully!");
179             } else {
180                 System.out.println("Reservation
update failed.");
181             }
182         }
183     } catch (SQLException e) {
184         e.printStackTrace();
185     }
186 }
187
188     private static void deleteReservation(Connection
connection, Scanner scanner) {
189         try {
190             System.out.print("Enter reservation ID
to delete: ");
191             int reservationId = scanner.nextInt();
192
193             if (!reservationExists(connection,
reservationId)) {
194                 System.out.println("Reservation not
found for the given ID.");
195                 return;
196             }
197
198             String sql = "DELETE FROM reservation
WHERE reservation_id = " + reservationId;
199
200             try (Statement statement = connection.
createStatement()) {
201                 int affectedRows = statement.
executeUpdate(sql);
```

```

202
203         if (affectedRows > 0) {
204             System.out.println("Reservation
deleted successfully!");
205         } else {
206             System.out.println("Reservation
deletion failed.");
207         }
208     }
209     } catch (SQLException e) {
210         e.printStackTrace();
211     }
212 }
213
214     private static boolean reservationExists(
Connection connection, int reservationId) {
215         try {
216             String sql = "SELECT reservation_id FROM
reservation WHERE reservation_id = " +
reservationId;
217
218             try (Statement statement = connection.
createStatement());
219                 ResultSet resultSet = statement.
executeQuery(sql)) {
220
221                 return resultSet.next(); // If there
's a result, the reservation exists
222             }
223         } catch (SQLException e) {
224             e.printStackTrace();
225             return false; // Handle database errors
as needed
226         }
227     }
228
229
230     public static void exit() throws
InterruptedException {
231         System.out.print("Exiting System");
232         int i = 5;

```



```
233         while(i!=0){
234             System.out.print(".");
235             Thread.sleep(455);
236             i--;
237         }
238         System.out.println();
239         System.out.println("ThankYou For Using Hotel
Reservation System!!!");
240     }
241 }
242
243
```