Summer - 2021

Report on Project

Online Hotel Reservation System

Course Code: CSE411 | **Section:** 02

Course Title: Software Engineering & Information System

Submitted By

Md Fahimul Islam (2017-1-62-022)

Md. Jusef (2017-2-60-160)

Wasiur Rahman (2017-1-60-115)

Submitted To

Dr. Mohammad Salah Uddin

Assistant Professor

Department of Computer Science & Engineering

Project Title:

Our Project name is "Online Hotel Reservation System".

Introduction:

Online Hotel Reservation system will provide service to online customers, and an administration. Online customer can make searches reservation and cancel an existing reservation on the hotel reservation's website. Administrator can add/update the hotel and room information.

The project, Online Hotel Reservation System is a web-based application that allows the hotel manager to handle all hotel activities online. Interactive GUI and the ability to manage various hotel bookings and rooms make this system very flexible and convenient. The hotel manager is a very busy person and does not have the time to sit and manage the entire activities manually on paper. This application gives him the power and flexibility to manage the entire system from a single online system. "Online hotel reservation" project provides room booking system. The system allows the manager to post available rooms in the system. Customers can view and book room online. Admin has the power of either approving or disapproving the customer's booking request. Other hotel services can also be viewed by the customers and can book them too. The system is hence useful for both customers and managers to portable manage the hotel activities.

Motivations

Cox's International Travel is an tourist agent which provides the facilities for booking hotels, houses, holiday apartments and other accommodations for customers. Any customers need to book hotel or apartment. they need to visit their office which is situated in Kolatoli for checking the availability as well as negotiation. Recently they have decided to change their booking pattern from manual to web based system. It will help them to manage the customers booking easily and also to keep the customers data safer. It will also help staffs to keep in track their customer's online booking request as well as easily to reply feedback to the customers. This is my motivation for work in this project. customer can easily book their choice able hotels or apartment.Saves time of customers for quickly reserving room.

Challenges

- Slot Management: Close Bookings
- Single Platform Multiple Booking Types
- Booking Cancellation Option for Customers
- Price Per Booking: Time Slot Management
- Manage Bookings via Calendar
- Streamline Online Booking Process
- Handle Customer Complaints
- User can create account and login.
- They can update their profile

Requirement Specification:

Functional Requirements:

Functional requirements are the fundamental actions that a system must perform. This system has many categories of functional requirements which are-

1.Registration:

The user must register into the system by entering name, contact information, Email Address. The email address and the contact information must be valid for authentication purpose. The user needs to give bank account or Bkash number for payment purpose which are also needed to be valid. During the registration the user must need to give a unique user id and a strong password for logging into the system in future. When the user will complete the registration, then the system should be able to store the registration information in the database. The system will also able to store the login information. The system should show a confirmation message to the user for confirming the registration. For invalid email address the system should not allow the user to register and will show an error message.

2. Login:

For login the user needs to input user id and password, then the system will check if both the inputs are correct or not if both are correct then a new page will open for the user with options available for further actions. In the case of invalid user id or password the system will show an error and the user can not be able to login and hast to give the user id and password again.

3. Booking/Reservation:

The user must reserve a room at least three days before their arrival in the hotel. User need to give all the information about the place where he/she wants to book a room. There should be an option including some popular location, so that the user can also choose the preferable location from there. If the user searches for a location which is not available or invalid in the system then, it will show a message like "Invalid location" then the user have to give the location again. After selecting the location user will get the names of the hotels available in that particular area. Then the user can choose the preferable hotel from there. The user also needs to give the information about how many days he/she wants to reserve/book the room.

4. Make Payment:

The payment system should be user friendly. The system should accept payment from both card and Bkash. The system will provide a bank account number if the user wants to give payment through card, or it will provide a Bkash number if the user wants to give payment through Bkash. After successful payment the system should show a message for confirming the user.

5. Cancel Reservation:

The user can cancel the reservation within two days of booking and then they will be refunded. If the user failed to cancel the reservation within two days, then there will not be refunded. In the same process the authority can also cancel the reservation, in that case the user will be refunded.

6. Admin Login:

The admin can login to the system by using the user id and password which are assigned to them. They can update the system and also can make any necessary changes in the system. All the actions should be recorded into the database of the system.

Non-Functional Requirements:

The non-functional requirements are the requirements which specifies the criteria that can be used to judge the operation of a system.

The non-functional requirements of our system are-

1. Performance Requirements:

The system should be functional for 24 hours a day to enable user interaction at any point of time.

2. Safety Requirements:

The database should be carefully maintained by the administrator any loss may harm the entire system.

3. Security Requirements:

The information of the user and the admins should be secured enough. The user id and password also needed to be encrypted. The database and the account information also needed to be secured properly. Any leakage of information can hamper the whole system.

4. Software Quality Attributes:

Adaptability: The system should be able to adapted with the new changes.

Availability: The system should be able to give services 24/7.

Interoperability: Each interface makes the use of information provided by other.

Flexibility: The system should provide highly flexible services to the users.

Portability: The system should run in every operating system such as Microsoft Windows, Ubuntu.

Maintainability: The project is going to be developed using php and MySQL so the system should be easy to maintain.

Feasibility Studies:

It is common practice to analyze the scope or idea of the project before taking any additional steps, and this is where the feasibility study comes into play. A comprehensive analysis of the practicality and sustainability of the software concept brings together all the elements of knowledge that help to realize whether it is worth investing in the project.

Technical feasibility:

The hotel booking case aims to bring a better experience for users who are busy with professional work and do not know the location and other information. It can provide a certain level of information. In order to develop this system we will make sure that the system will give an easy and quick access to the users. Hotel managers can frequently update the database and publish available rooms in the system. This application is developed using HTML, PHP, CSS and MYSQL and more within given time.
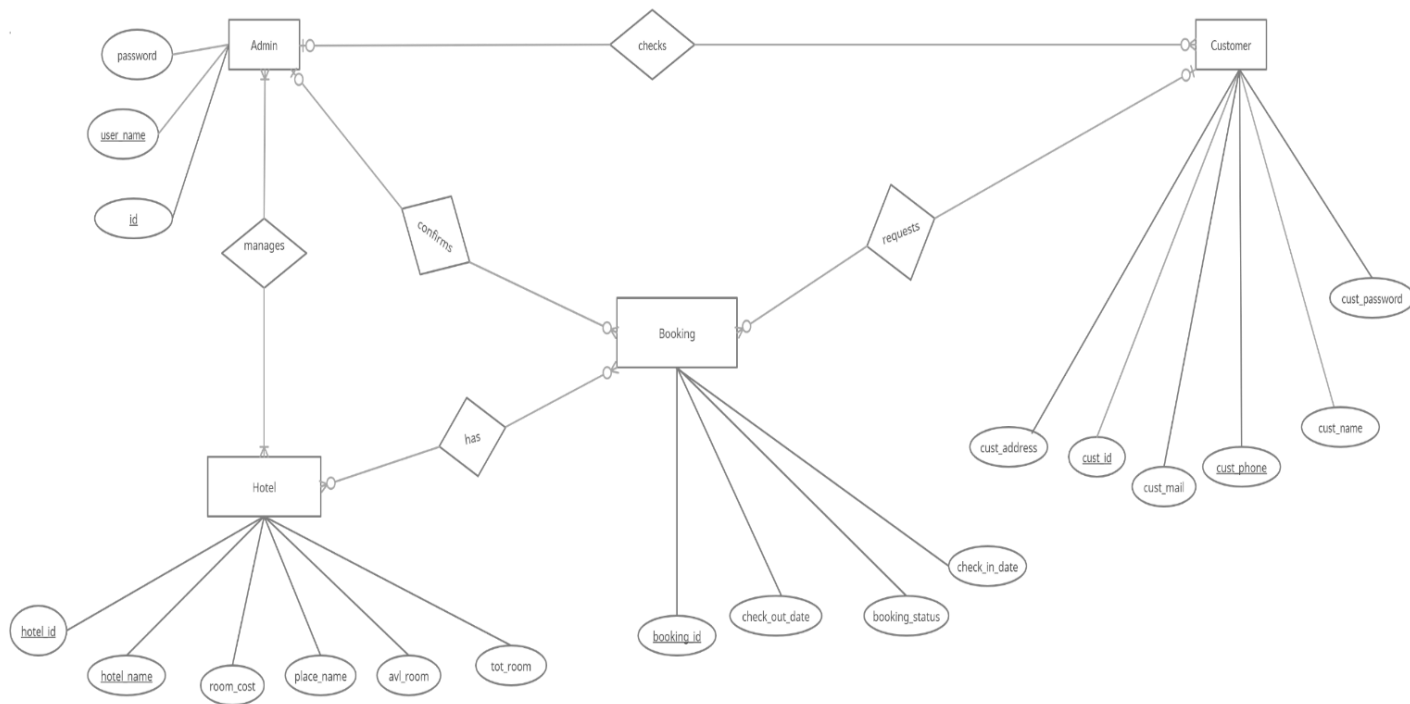
Operational feasibility:

Since we live downtown, finding available hotel reservations for the desired route is also a problem. It is difficult to find a decent room that meets all user requirements. so our system is here

to help people find better rooms easily. We prepare facilities for booking hotels, houses, holiday apartments and other accommodations to our customers. Customers can easily book the room they want. The owner can confirm the reservation and offer the room within the delivery time. Customers can pay after booking online. This system is very useful for those who need a room or apartment in many cases.
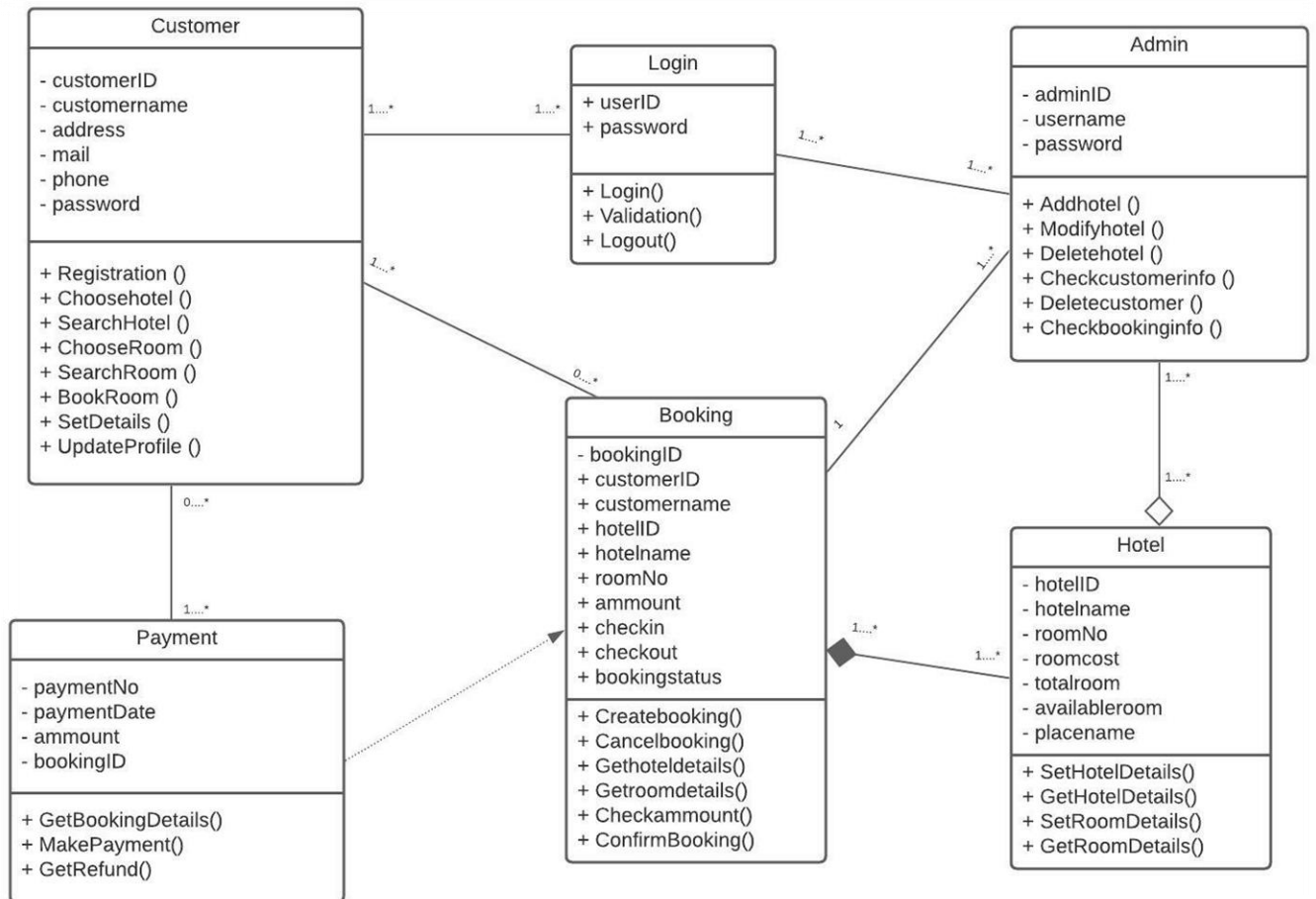
Economic Feasibility:

This project is not profitable, because we no longer consider making money from it, so there is no economic feasibility.
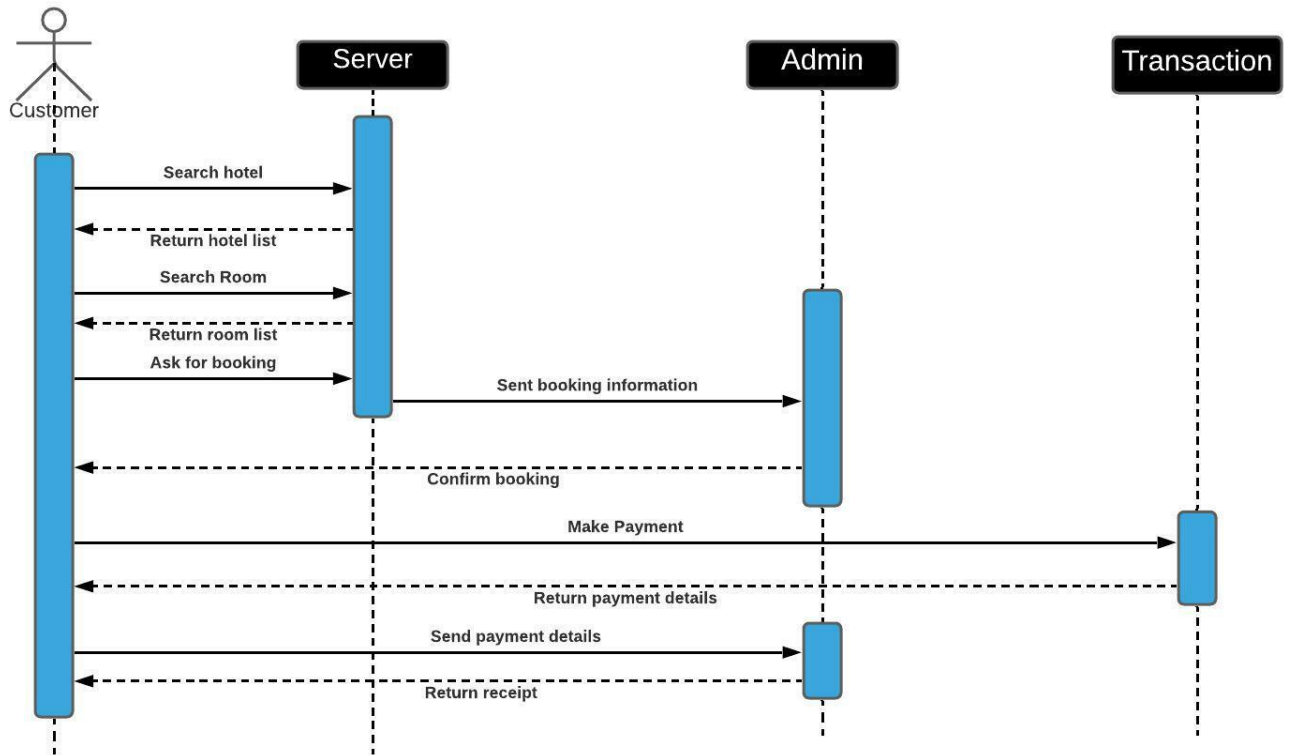
ER diagram

UML Class Diagram:

Customer

- customerID
- customername
- address
- mail
- phone
- password

+ Registration ()
+ Choosehotel ()
+ SearchHotel ()
+ ChooseRoom ()
+ SearchRoom ()
+ BookRoom ()
+ SetDetails ()
+ UpdateProfile ()

Login

+ userID
+ password

+ Login()
+ Validation()
+ Logout()

Admin

- adminID
- username
- password

+ Addhotel ()
+ Modifyhotel ()
+ Deletehotel ()
+ Checkcustomerinfo ()
+ Deletecustomer ()
+ Checkbookinginfo ()

Booking

- bookingID
+ customerID
+ customername
+ hotelID
+ hotelname
+ roomNo
+ ammount
+ checkin
+ checkout
+ bookingstatus

+ Createbooking()
+ Cancelbooking()
+ Gethoteldetails()
+ Getroomdetails()
+ Checkammount()
+ ConfirmBooking()

Payment

- paymentNo
- paymentDate
- ammount
- bookingID

+ GetBookingDetails()
+ MakePayment()
+ GetRefund()

Hotel

- hotelID
- hotelname
- roomNo
- roomcost
- totalroom
- availableroom
- placename

+ SetHotelDetails()
+ GetHotelDetails()
+ SetRoomDetails()
+ GetRoomDetails()

Relationship multiplicities: 1....*, 1....*, 1....*, 1....*, 0....*, 1....*, 1....*, 1, 0....*, 1....*, 1....*, 1....*

Sequence Diagram

Customer	Server	Admin	Transaction

Search hotel

Return hotel list

Search Room

Return room list

Ask for booking

Sent booking information

Confirm booking

Make Payment

Return payment details

Send payment details

Return receipt

Activity Diagram:

Use Case Diagram:
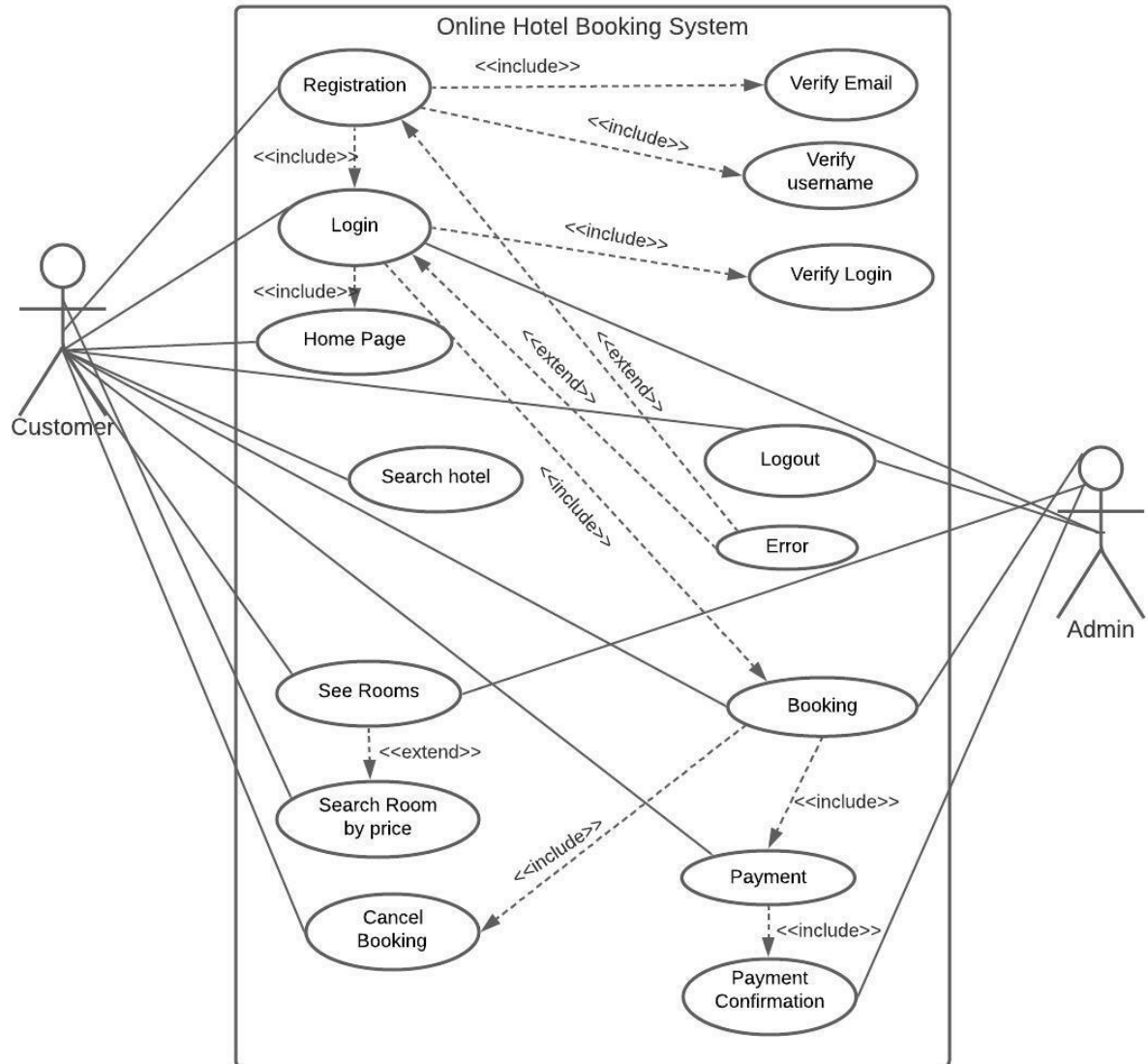
Use Case Description:

1

Use case #1	Register	
Goal in context	This use case helps customer to create account in this webpage	
Preconditions	Customers must have valid mail and unique username	
Success end Condition	Customers will be able to create accounts in this webpage	
Failed end Condition	Try again	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer must browse the website	
Description	**Step**	**Action**
	1	Need to provide a unique username, valid email and password, confirm password and click register button
Extensions or variations	**Step**	**Branching Action**
	1	Username already exists error message

Use case #2	Login	
Goal in context	This use case helps customer to login into the website using their existing account	
Preconditions	Customers must have their registered account. The website should be running.	
Success end Condition	After successful login customers can see their profile and other features	
Failed end Condition	Try again	
Primary actor	Customer, Admin	
Secondary actor	Hotel	
Trigger	Customer must register into the website	
Description	**Step**	**Action**
	1	Need to enter their own username and password then have to click login button
Extensions or variations	**Step**	**Branching Action**
	1	Incorrect username or password error message

Use case #3	Homepage	
Goal in context	This use case helps customer to see the homepage and features shortcut	
Preconditions	Customers must browse this website	
Success end Condition	Customers can see all the features	
Failed end Condition	Try again	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer must browse the website	
Description	**Step**	**Action**
	1	Customer need to browse the website by clicking website link
Extensions or variations	**Step**	**Branching Action**
	1	Connection error message

Use case #4	Verify Email	
Goal in context	This use case checks if the customer provides a valid email or not when they create an account in this website	
Preconditions	Customers must have valid email id	
Success end Condition	After verification, customers can continue their registration	
Failed end Condition	Try again	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer try to register	
Description	**Step**	**Action**
	1	Customer need to provide a valid email
Extensions or variations	**Step**	**Branching Action**
	1	Invalid email error message

Use case #5	Verify Username	
Goal in context	This use case will verify the username when customer create an account	
Preconditions	Customer must have valid username	
Success end Condition	After verification, customer can continue their registration	
Failed end Condition	Try again	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer try to register	
Description	**Step**	**Action**
	1	Need to enter a valid username
Extensions or variations	**Step**	**Branching Action**
	1	Username already exists error message

Use case #6	Verify Login	
Goal in context	This use case helps customer to verify login	
Preconditions	Customers must have an account	
Success end Condition	After verification, user can continue their login	
Failed end Condition	Try again	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer try to login	
Description	**Step**	**Action**
	1	Customers need to enter username and password then need to click the login button
Extensions or variations	**Step**	**Branching Action**
	1	Incorrect username and password error message

Use case #7	Search hotel	
Goal in context	This use case helps customer to search for hotels	
Preconditions	Customers must have an account	
Success end Condition	After successful search, customers can see their desired hotels	
Failed end Condition	Try again	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer must login first	
Description	**Step**	**Action**
	1	Need to select hotel name from the list

Use case #8	Error	
Goal in context	This use case helps to identify error	
Preconditions	Customers must try to register or login	
Success end Condition	Shows error message	
Failed end Condition	Does not show any error message	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customers try to register or login	
Description	**Step**	**Action**
	1	Customers need to try to register or login on this website
Extensions or variations	**Step**	**Branching Action**
	1	Customers must try performing programs or error messages will not show

Use case #9	See Rooms	
Goal in context	This use case helps customer to watch available rooms	
Preconditions	Customers must have an account	
Success end Condition	After watching room customers can be able to choose their desired room	
Failed end Condition	Try again	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer must login first	
Description	**Step**	**Action**
	1	Need to choose room from the list

Use case #10	Search room by price	
Goal in context	This use case helps customer to search rooms by price	
Preconditions	Customers must have an account	
Success end Condition	After successful search customer can see the rooms for the price range they entered	
Failed end Condition	Try again	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer must login first	
Description	**Step**	**Action**
Extensions or variations	**Step**	**Branching Action**
	1	Username already exists error message

Use case #11	Booking	
Goal in context	This use case helps customer to book hotel room	
Preconditions	Customers must have an account	
Success end Condition	Customers can book rooms	
Failed end Condition	Try again	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer must login first	
Description	**Step**	**Action**
	1	Click on book button

Use case #12	Payment	
Goal in context	This use case helps customer to make payment	
Preconditions	Customers must have an account	
Success end Condition	Customers will be able to make payment according to the total ammount	
Failed end Condition	Try again	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer must login first	
Description	Step	Action
	1	Customers need to provide all the information correctly

Use case #13	Payment Confirmation	
Goal in context	This use case helps customer to confirm their payment	
Preconditions	Customers must have an account	
Success end Condition	Customers will be able to confirm their payment	
Failed end Condition	Try again	
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer must login first	
Description	Step	Action
	1	Need to click confirm button

Use case #14	Cancel Booking	
Goal in context	This use case helps customer to cancel booking	
Preconditions	Customers must have an account	
Success end Condition	Customers will be able to cancel their booking within a valid time	
Failed end Condition		
Primary actor	Customer	
Secondary actor	Admin	
Trigger	Customer must login first	
Description	Step	Action
	1	Need to click on the Cancel Booking button
Extensions or variations	Step	Branching Action
	1	Cancelation time exceeded error message

Use case #15	Logout	
Goal in context	This use case helps customer and admin to log out from the website	
Preconditions	Customers must have valid mail and unique username	
Success end Condition	After doing logout customer and admin will get logged out from the website but the can see the website in general view	
Failed end Condition		
Primary actor	Customer, Admin	
Secondary actor		
Trigger	Customer and admin must login first	
Description	**Step**	**Action**
	1	Click on logout button
Extensions or variations	**Step**	**Branching Action**
	1	The button will not be shown without logging in

Software Testing

Test Code for show customer:

```php
<!DOCTYPE html>
<?php
    include('../connection.php');
?>
<?php include('partials/menu.php'); ?>
<html>
    <body>
        <h1>User Details</h1>
        <style>
        table {
        border-collapse: collapse;
         }

         table, th, td {
         border: 1px solid black;
          }
        </style>
        <table class = "table table-border">
            <tr>
                <th>customer_name</th>
                <th>customer_id</th>
                <th>customer_password</th>
                <th>customer_address</th>
                <th>customer_mail</th>
                <th>customer_phone</th>
            </tr>
            <?php
                $sql = "SELECT * FROM customer";
                $result = $conn->query($sql);
                if($result->num_rows>0)
                {
                    while($row = $result->fetch_assoc())
                    {
            ?>
            <tr>
                <td><?php echo $row['cust_name']; ?></td>
                <td><?php echo $row['cust_id']; ?></td>
                <td><?php echo $row['cust_password']; ?></td>
```

show_customer.php ●

Online Hotel Booking > admin > show_customer.php

Ln 47, Col 17   Spaces: 4   UTF-8   CRLF   PHP

29°C Haze   10:43 PM   27-Aug-21

```php
                    while($row = $result->fetch_assoc())
                    {
        ?>
        <tr>
            <td><?php echo $row['cust_name']; ?></td>
            <td><?php echo $row['cust_id']; ?></td>
            <td><?php echo $row['cust_password']; ?></td>
            <td><?php echo $row['cust_address']; ?></td>
            <td><?php echo $row['cust_mail']; ?></td>
            <td><?php echo $row['cust_phone']; ?></td>
        </tr>
        <?php
    }
    }
    ?>
    </table>
    <button><a class="text-decoration:none" href="index.php">return to main menu</a></button>
</body>
</html>
```

29°C Haze   10:46 PM
27-Aug-21

White box testing solution:
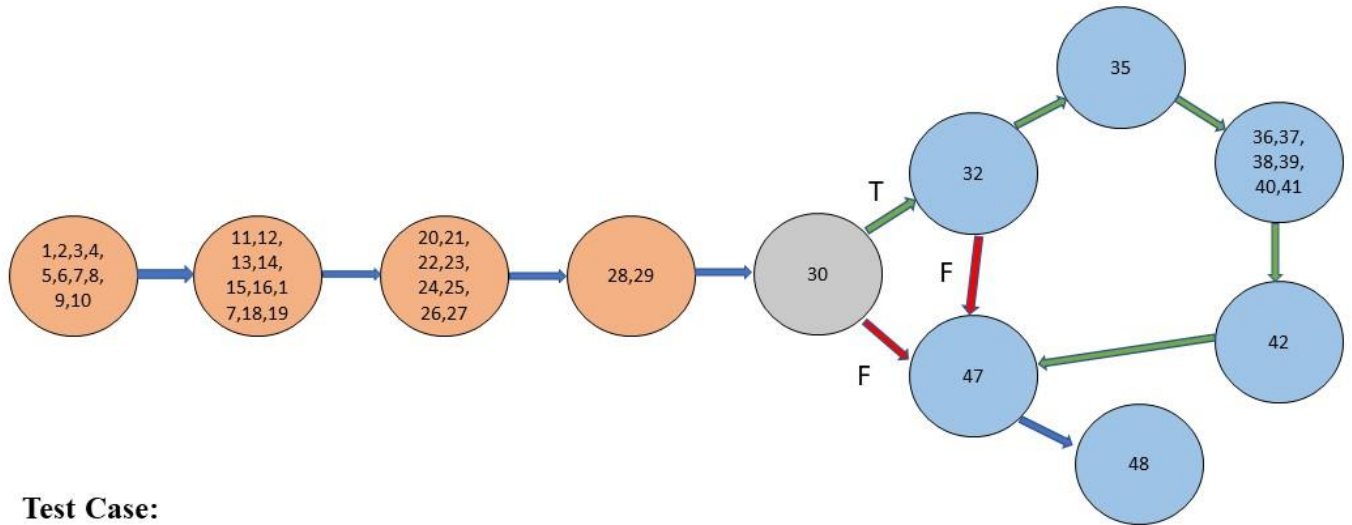
Test Case:
1: If there any matching result in the server.
2: If there are not matching result in the server.

Test Code for show Hotel:

```php
1   <!DOCTYPE html>
2   <?php
3       include('../connection.php');
4       include('partials/menu.php');
5   ?>
6   <html>
7       <body>
8           <h1 class="text-center">Hotel Details</h1>
9           <style>
10          table {
11          border-collapse: collapse;
12           }
13           table, th, td {
14           border: 1px solid black;
15            }
16          </style>
17          <table class = "table table-border col-10 offset-1">
18              <tr>
19                  <th>Hotel id</th>
20                  <th>Hoel name</th>
21                  <th>Place Name</th>
22                  <th>Room Cost</th>
23                  <th>Total Room</th>
24                  <th>Available Room</th>
25              </tr>
26              <?php
27                  $sql = "SELECT * FROM hotel";
28                  $result = $conn->query($sql);
29                  if($result->num_rows>0)
30                  {
31                      while($row = $result->fetch_assoc())
32                      {
33              ?>
34              <tr>
35                  <td><?php echo $row['hotel_id']; ?></td>
36                  <td><?php echo $row['hotel_name']; ?></td>
37                  <td><?php echo $row['place_name']; ?></td>
38                  <td><?php echo $row['room_cost']; ?></td>
```
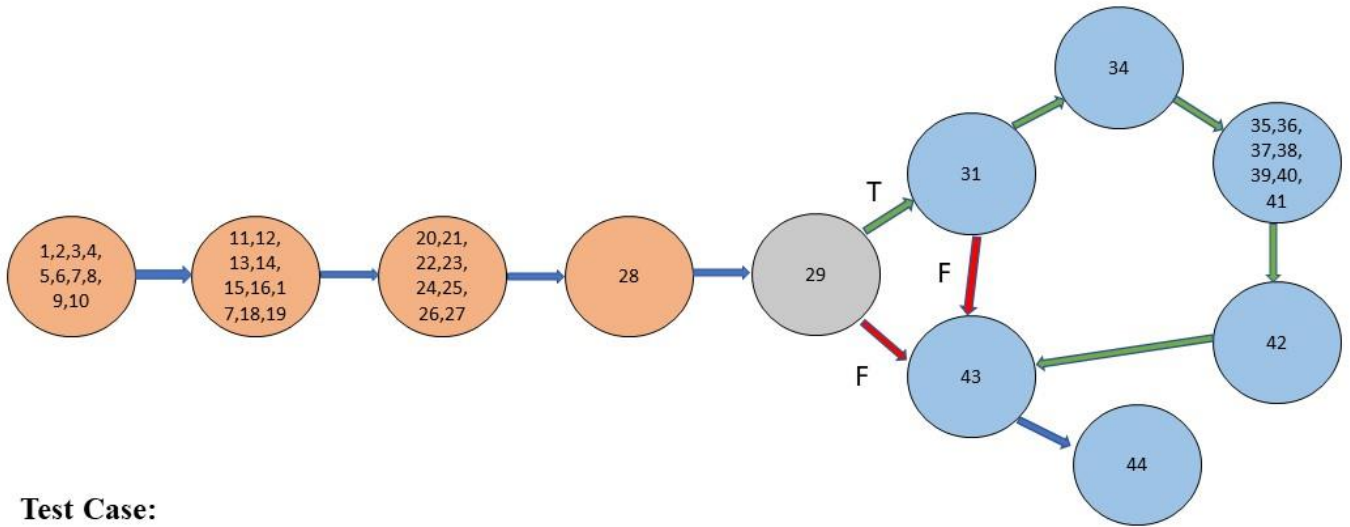
Ln 46, Col 8   Spaces: 4   UTF-8   CRLF   PHP

29°C Haze   11:05 PM 27-Aug-21

```php
                {
        ?>
        <tr>
            <td><?php echo $row['hotel_id']; ?></td>
            <td><?php echo $row['hotel_name']; ?></td>
            <td><?php echo $row['place_name']; ?></td>
            <td><?php echo $row['room_cost']; ?></td>
            <td><?php echo $row['tot_room']; ?></td>
            <td><?php echo $row['avl_room']; ?></td>
        </tr>
        <?php } } ?>
    </table>
    <button class="btn btn-primary col-2 offset-1 text-center"style="color:black"><a class="text-decoration:none" href="index.php">re
</body>
</html>
```

White box testing solution:

Test Case:
1: If there any matching result in the server.
2: If there are not matching result in the server.

Future work:

1. Suitable Details Room suggestion.
2. Interactive User Interface.
3. Developing an App.
4. Integrating more features.
5. High level Security provide.