Group Experiment

Name of the Student:-

Preksha Patel - 60004210126

Anushka Pandit - 60004210119

Khushi Jobanputra - 60004210147

Varenya Uchil - 60004210121

Shashwat Shah - 60004220126

SOFTWARE TESTING AND QUALITY ASSURANCE (STQA)

EXPERIMENT 5

Aim: To design test cases for a given problem statement based on Decision Table Testing.

Scenario/Use Case for Condition Testing:

Use Case Description: Book Hotel Room via Online Portal

Actor: Website User

- This could be any individual looking to book a hotel room via a website. The user could be a traveler looking for accommodations for business or leisure, a group booking rooms for a vacation, or someone needing a last-minute booking.

Goal: Book a hotel room based on specific preferences and requirements.

- The user's primary objective is to find and book a hotel room based on location, dates, room type, and price range. The user may also want to view room details, availability, and make a secure payment.

Detailed Steps:

1. **User Discovery:** The user learns about the website's booking functionality, either through direct access, recommendation, or online search.
2. **Search Criteria Location:** The user identifies the search bar or filter options on the website to start the search. This can include entering location, check-in/check-out dates, and the number of guests.
3. **Search Input:** The user enters the required data (e.g., destination, dates, number of rooms, guests) and clicks the "Search" button.

4. **Search Initiation:** The user initiates the search, triggering the website to fetch available hotels that meet the criteria.
5. **Search Processing:** The website communicates with a backend system (potentially an API) to retrieve available hotels and their details.
6. **Search Result Display:** The website processes the retrieved hotel information and displays the results. The user can view the details of each hotel, such as pricing, ratings, and room types available.
7. **Booking Process:** The user selects a hotel, picks a room, and proceeds to the booking page for payment and final confirmation.

Additional Considerations:

- The website might offer additional functionalities to filter results, such as sorting by price, star rating, and guest reviews.
- The website should handle unexpected inputs gracefully, such as invalid dates or a non-existent location.
- Error messages should guide the user in case of incorrect inputs or unavailable rooms.

Conditions for Testing:

The conditions for testing are derived from the scenario and functionalities involved:

- **Location Search:**
  - Valid Location (A city or region where hotels are available)
  - Non-existent Location (A city or region where no hotels are available)
  - Invalid Location (Special characters, empty string)
- **Room Availability:**
  - Rooms Available (Hotels with rooms matching the search criteria)
  - No Rooms Available (No rooms match the search criteria)
- **Payment Process:**
  - Valid Payment Method (Valid credit/debit card or payment gateway)
  - Invalid Payment Method (Expired credit card or invalid payment details)

Test Cases:

Based on the decision table, here are some test cases:

1. **Test Case 1:**
   - **Condition:** Location Search - Valid Location, Room Availability - Rooms Available, Payment Method - Valid Payment
   - **Test Steps:**
     - Search for a location with available hotels.
     - Verify that rooms matching the search criteria are displayed.
     - Select a room and proceed to payment. Verify that the payment page is accessible and the payment is successful.
2. **Test Case 2:**
   - **Condition:** Location Search - Valid Location, Room Availability - No Rooms Available, Payment Method - Not Applicable

- **Test Steps:**
    - Search for a location with no available rooms.
    - Verify an appropriate message is displayed indicating that no rooms are available.

3. **Test Case 3:**
    - **Condition:** Location Search - Invalid Location (e.g., special characters or empty string), Room Availability - Not Applicable, Payment Method - Not Applicable
    - **Test Steps:**
        - Search for an invalid location (e.g., empty string or special characters).
        - Verify that an appropriate error message is displayed indicating invalid input.
4. **Test Case 4:**
    - **Condition:** Location Search - Non-existent Location, Room Availability - Not Applicable, Payment Method - Not Applicable
    - **Test Steps:**
        - Search for a non-existent location (e.g., a city that doesn't have hotels listed on the website).
        - Verify that an appropriate message is displayed indicating no hotels found for the location.
5. **Test Case 5:**
    - **Condition:** Location Search - Valid Location, Room Availability - Rooms Available, Payment Method - Invalid Payment
    - **Test Steps:**
        - Search for a location with available hotels.
        - Select a room, and proceed to the payment page.
        - Enter invalid payment details (e.g., expired credit card) and verify that an appropriate error message is displayed.

Decision Table for the above scenario:

Conditions	**Test Case 1**	**Test Case 2**	**Test Case 3**	**Test Case 4**	**Test Case 5**
Location Search	Valid Location	Valid Location	Invalid Location	Non-existent Location	Valid Location
Room Availability	Rooms Available	No Rooms Available	Not Applicable	Not Applicable	Rooms Available
Payment Method	Valid Payment	Not Applicable	Not Applicable	Not Applicable	Invalid Payment
Room Display	Rooms Listed	No Rooms Available	Error Message Display	No Rooms Available	Error Message Display
Navigation	Booking Successful	No Booking Available	Error Message Display	No Booking Available	Error Message Display

Rules for Valid and Invalid Input/Output Conditions:

- **Valid Input:**

    - **Location Search:** Valid location name (city, region).
    - **Room Availability:** Rooms matching the search criteria are displayed.
    - **Payment:** Valid payment method (credit/debit card).
    - **Navigation:** Successful transition to booking page, followed by successful payment.
- **Invalid Input:**
    - **Location Search:** Non-existent or invalid location.
    - **Room Availability:** No rooms match the search criteria.
    - **Payment:** Invalid payment method (e.g., expired card or incorrect details).
    - **Navigation:** Failure to transition to the booking or payment page due to invalid input.

Observations and Technical Problems Noticed:

Observations:

1. **Room Availability Issues:** Sometimes the available rooms are not displayed correctly, or rooms might not match the search criteria.
2. **Search Performance:** The website may experience slow response times, especially with multiple filters applied.
3. **Payment Gateway Issues:** There may be issues with specific payment gateways (e.g., cards being rejected or payments failing due to integration bugs).

Technical Problems:

1. **Data Synchronization Issue:** The website may fail to retrieve up-to-date room availability, showing rooms as available when they are actually booked.
2. **Payment System Errors:** The payment processing system might not always validate cards properly, leading to false rejections.
3. **UI/UX Problems:** The website's interface might not update dynamically when users change search filters, leading to incorrect results being shown.

Conclusion:

Based on the test case results, conclusions can be drawn about the website's functionality. Did it meet expectations for booking accuracy, error handling, and payment processing? Were there any areas requiring improvement, such as faster response times, better search result accuracy, or improved navigation?

This decision table testing ensures that the hotel room booking process is thoroughly tested, accounting for various scenarios and edge cases.

References:

- https://testsigma.com/blog/decision-table-testing/
- https://www.geeksforgeeks.org/decision-table-based-testing-in-software-testing/
- https://www.javatpoint.com/decision-table-technique-in-black-box-testing