Assessment – Manual Testing

Below is the test data created that will be used for TC execution on 26-06-2023

Test Data					
Place	Floor	Room	Smoking Allowed	Equipment	Available from
WP1	1	5	[]	PC, Phone, Projector	24-06-2023 to 30-06-2023
WP2	1	10	[x]	PC, Flip Chart Board	28-06-2023 to 12-07-2023
WP3	2	9	[x]	PC	23-06-2023 to 23-06-2023
WP4	3	1	[x]	Phone, Flip Chart Board	01-05-2023 to 28-08-2023
WP5	3	6	[]		25-06-2023 to 28-06-2023
WP6	4	4	[]	PC, Phone, Projector, Flip Chart Board	24-06-2023 to 25-06-2023
WP7	5	8	[x]	Phone	27-07-2023 to 30-09-2023
WP8	6	2	[]	Projector, PC	02-01-2023 to 06-04-2023
WP9	12	7	[x]	Flip Chart Board	06-06-2023 to 07-06-2023

Task 1: Requirements Analysis and Coverage

Scenario 1: Validate the UI elements

Expected Results:

1. User should be able to view that all the visible elements are properly aligned

2. User should be able to view that date fields like From/To have calendar box option to select date

3. User should be able to see a checkbox next to 'Smoking Allowed' label

4. User should be able to see and select from a list of items next to 'Equipment' label and Scroll bar should work properly

5. User should be able to see a Search button that should be clickable

6. User should be able to see a result box containing information based on the search criteria

7. Search Result Box will have columns: Place, Floor, Room, Smoking Allowed, and Equipment

8. User should be able to see page number navigation based on search results

9. Page should not break in any case. (Instances: While inputting the data, while searching, while navigating)

Scenario 2: Verify input data for all the fields

Expected Results:

1. From/To - User should be able to select date range from calendar box and date cannot be past dated. (Assumption: If manual date input is accepted then user should be able to enter date manually as per valid date format)

2. Floor - User should be able to enter numeric value.

Error message should be displayed in case user enters decimal value/negative value/character/special character.

(Assumption: If multiple comma separated values are accepted user should be able to input that.)

3. Smoking allowed - User should be able to click on checkbox.

(Assumption: Default value is unchecked)

4. Equipments - User should be able to select multiple values from list.

(Assumption: Default value is unchecked for all the values)

Scenario 3: Verify user is able to see workplaces in the result box based on search criteria
Expected Results:
1. User should be able to see results based on filter applied.
(We can use test data to execute different scenario.)
2. Based on the search results if pagination configuration is exceeded then user should be able to see page navigation options.
3. Result box should be read only.

Scenario 4: Verify user is able to search without selecting any criteria
Expected Results:
1. User should be able to see search result for all the available workplaces.
(Assumption: Result box will show all the workplace available on or from the present day)

Scenario 5: Verify user is able to sort the search results
Expected Results:
1. User should be able to sort the search results by clicking on sort arrow which is available next to all the headers available in result box.

Scenario 6: Verify response time of application
Expected Results:
1. User should be able to search the workplaces within the stipulated time.
(Assumptions: Results should appear within 10000ms)
2. If due to any network/data issue response is taking more than usual time then user should be displayed a loading message intimating about the delay while searching.

Task 2: Estimation

Estimated Test Activities with WBS (Work Breakdown Structure):

Assumptions:

The requirements and user stories provided are complete and well-defined.

Estimation of each user story will be given points on the basis of complexity of feature.

Sufficient resources and test environments will be available for testing.

The development team will deliver the features as per commitments.

Any Blocker identified is reported well in time so appropriate action can be taken.

Test Planning

1.1. Review requirements and user stories

1.2. Identify test objectives and scope

1.3. Define test strategy and approach

1.4. Identify test deliverables and milestones

1.5. Produce a test plan document covering above mentioned details

Test Environment Setup

2.1. Identify hardware and software requirements

2.2. Set up all the necessary equipments/application for test environment

2.3. Install and configure the application

2.4. Prepare test data and test accounts

User Registration Testing

3.1. Verify user registration process (Assuming: New user will have to sign up to login)

3.2. Validate user account creation and login functionality

3.3. Verify user profile management features

Office Place Booking Testing

4.1. Test the ability to view available working places

4.2. Verify the booking functionality for current and future dates

4.3. Validate the prevention of double booking (Assuming: Redundant bookings should not be allowed)

4.4. Test the cancellation of bookings and validating that workplace is available for booking to other users.

User Roles Testing

5.1. Test user role assignment and management

5.2. Validate the access control based on user roles

5.3. Test role-specific functionalities for users and office managers

Database Testing

6.1. Test all the relevant tables are storing correct information

6.2. Validate checks for data redundancy

Native Mobile Application Testing

7.1. Verify the installation and setup of the mobile application

7.2. Verify application works seamlessly on different operation system like IOS/Android etc

7.3. Verify user registration and profile management on mobile application

7.4. Verify user is able to search and view results

7.5. Test the functionality of booking and cancelling working places

7.6. Validate usability and accessibility of application

Office Place Management Testing

8.1. Test the creation of new working places by office managers

8.2. Verify the ability to update and delete working places

8.3. Validate the accuracy and up-to-date information display

Test Execution

9.1. Execute test cases for each feature

9.2. Record test results and any bugs/defects found

9.3. Perform integration testing

9.4. Perform regression testing

9.5. Verify fixes for reported bugs/defects

9.6. Conduct retesting after fixes

Test Reporting and Documentation

10.1. Generate test reports summarizing test execution and results

10.2. Document any issues, risks, and recommendations

10.3. Update test documentation and test cases

Customer Deliverables:

- Test Plan document outlining the test objectives, scope, and approach.
- Test Environment setup instructions for the required hardware and software.
- Test cases and test data for each user story.
- Test execution reports detailing the test results and any defects found.
- Updated test documentation and test cases.

High-Level Description of Estimated Test Activities Summed Up in a Process:

The estimated test activities follow a structured approach to ensure the quality and reliability of the new features added to the application.

Test planning involves reviewing the requirements and user stories, defining objectives and scope, and creating a test plan document to guide the testing process.

Test environment setup focuses on identifying the necessary hardware and software requirements, configuring the test environment, and preparing the application for testing.

Test case design involves identifying test scenarios based on user stories and creating test cases to cover different functionalities. The test cases are prioritized based on risk and importance and undergo a review process.

Test execution is the phase where the test cases are executed, results are recorded, and any defects found are reported. Regression testing is performed to ensure that existing functionality is not affected, and fixes for reported defects are verified.

Test reporting and documentation include generating test reports summarizing test execution and results, documenting issues, risks, and recommendations, and updating test documentation and test cases based on the testing outcomes.