* **Task 1: Requirements Analysis and Coverage**

**Test Plan: Workplace Search Functionality**

**1. Introduction**

This document outlines the testing approach for verifying the functionality of the workplace search feature in the organization's web application. The testing will ensure that the system meets the specified requirements and provides an efficient and user-friendly experience.

**2. Objectives**

Validate the search functionality according to the provided requirements.

Ensure accurate sorting and navigation of search results.

Assess the system's performance under various conditions.

Verify the user interface for usability and readability.

**3. Scope**

**The testing will focus on:**

Searching for workplaces based on date, floor, equipment, and smoking restrictions.

Sorting and navigating through search results.

Validating user input before executing the search.

Evaluating the responsiveness and user-friendliness of the interface.

**4. Test Data**

**4.1 Valid Data**

**Date:** Current or future dates

**Floor:** Existing floor numbers

**Equipment:** Valid combinations

**Smoking Restrictions:** "Smoking" or "Non-smoking"

**4.2 Invalid Data**

**Date:** Past dates

**Floor:** Non-existent floor numbers

**Equipment:** Invalid combinations

**Smoking Restrictions:** Invalid values

**4.3 Boundary Data**

**Date:** Earliest and latest possible dates

**Floor:** Lowest and highest floor numbers

**Equipment:** Minimum and maximum numbers of equipment

**4.4 Extreme Data**

Large datasets

Empty search criteria

Unusually long strings

**4.5 Performance Data**

Large datasets for response time measurement

Stress testing with concurrent searches

**5. Test Cases:**

* **Search Functionality Testing :**

**Test Case 1:** Verify search without setting parameters.

**Test Case 2:** Verify search with provided parameters.

**Test Case 3:** Verify search restricted to future dates.

**Test Case 4:** Verify data validation before search.

**Test Case 5:** Verify timely search results.

* **Sorting and Navigation Testing :**

**Test Case 6:** Verify sorting by all available fields.

**Test Case 7:** Verify flexible navigation between results.

* **User Interface Testing :**

**Test Case 8:** Verify user-friendly interface.

**Test Case 9:** Verify read-only search result table.

* **Boundary Testing :**

**Test Case 10:** Verify handling of lowest and highest floor numbers.

**Test Case 11:** Verify handling of earliest and latest dates.

**Test Case 12:** Verify handling of minimum and maximum equipment.

* **Performance Testing :**

**Test Case 13:** Measure response time with large datasets.

**Test Case 14:** Verify system performance under stress.

**6. Test Environment:**

**Web browser:** Chrome, Firefox, Safari

**Operating Systems:** Windows, macOS, Linux

**Devices:** Desktop, Laptop, Tablet, Mobile

**7. Test Execution:**

Test cases will be executed manually.

**8. Deliverables**

Test reports summarizing results and any issues encountered.

Recommendations for improvements if necessary.

**9. Risks and Assumptions:**

Assumption: The development environment closely mirrors the production environment.

Risk: Potential delays due to unforeseen technical issues or system constraints.

**10. Sign-off:**

Testing completion will require sign-off from stakeholders.

**Task 2 – Estimation**

**Test Activities Estimation**

**1. User Registration**

* Test Planning and Preparation
* Define test strategy and approach for user registration feature.
* Prepare test environment.
* Test Execution
* Verify registration process with valid data.
* Verify registration process with invalid data.
* Verify registration process with boundary data.
* Defect Reporting and Management
* Document and report any issues found during testing.
* Verify fixes and retest.

**2. Booking of Office Place**

* Test Planning and Preparation
* Define test strategy and approach for booking office place feature.
* Prepare test environment.
* Test Execution
* Verify booking process with valid data.
* Verify booking process with invalid data.
* Verify booking process with boundary data.
* Defect Reporting and Management
* Document and report any issues found during testing.
* Verify fixes and retest.

**3. User Roles (User and Office Manager Roles)**

* Test Planning and Preparation
* Define test strategy and approach for user roles feature.
* Prepare test environment.
* Test Execution
* Verify user role functionality (user and office manager) with valid data.
* Verify user role functionality with invalid data.
* Verify user role functionality with boundary data.
* Defect Reporting and Management
* Document and report any issues found during testing.
* Verify fixes and retest.

**4. Native Mobile Application**

* Test Planning and Preparation
* Define test strategy and approach for mobile application testing.
* Prepare mobile testing environment.
* Test Execution
* Verify mobile app functionality for all features.
* Test mobile app performance and usability.
* Defect Reporting and Management
* Document and report any issues found during mobile app testing.
* Verify fixes and retest.
* Process Overview

**Test Planning and Preparation:** Establishing test strategy, environment setup, and test case creation.

**Test Execution:** Actively testing the features according to the defined test cases.

**Defect Reporting and Management**: Documenting, reporting, and tracking issues found during testing.

**Assumptions:**

Adequate documentation and requirements understanding are available.

Testing will be done manually unless automation is explicitly mentioned.

Test environment setup and access to necessary resources are readily available.

Development and testing timelines are aligned.

**Expected Result:**

The customer receives a thoroughly tested increment of the application with all the new features integrated seamlessly. Quality assurance ensures that the new features work as expected, are user-friendly, and adhere to the defined requirements and standards.

**Task 3 – Complex Reporting**

* **Evaluation Report**

**Project Overview:**

We tested some software skills through different tasks. We looked at how well someone could understand requirements, estimate work, and test software features.

**Task 1 - Requirements and Testing:**

We checked how well someone could understand what the software needs to do. Then, we made test cases to make sure the software works correctly.

**Task 2 - Estimation:**

We estimated how much time and effort it would take to test new features in the software. We broke down the work into smaller parts and made guesses about how long each part would take.

**Task 3 - Reporting:**

We wrote this report to show what we did and what we found. We explained everything clearly so that anyone reading it would understand.

**Conclusion:**

The tasks showed good skills in understanding requirements, estimating work, and reporting findings. We suggested practicing more and looking for ways to work more efficiently.

**Recommendations:**

Keep practicing and look for ways to improve testing skills. Also, consider using automation to make testing faster and easier.