## **Test Plan: Resource Management**

### **1. Introduction**

This test plan outlines the testing approach for managing resources, services, and Book appointments in the Appointment Module. It includes the testing objectives, scope, test deliverables, and the test environment.

#### **1.1 Objectives**

* To verify the functionality of adding, editing, viewing, and deleting services.
* To ensure accurate specification of resource availability and associated services.
* To verify the functionality of adding, editing, viewing, and deleting resources.
* To ensure accurate specification of resource availability and associated resources.
* To validate the functionality of scheduling appointments.
* To ensure accurate processing and management of appointments.
* To verify integration with other modules (if applicable)

#### **1.2 Scope**

The test plan covers the following aspects:

* Adding resources.
* Editing existing resources.
* Viewing resource details.
* Deleting resources.
* We are specifying resource availability and associated services.
* Adding services.
* Editing existing services.
* Viewing service details.
* Deleting services.
* Scheduling appointments.
* Managing appointment details.
* Handling appointment conflicts.
* Integration with other modules (if applicable)

#### **1.3 Test Deliverables**

The following deliverables will be produced as part of this test plan:

* Test cases for resource, service, and appointment management.
* Test execution reports.
* Defect reports, if any.

#### **1.4 Test Environment**

The test environment for resource management includes:

* The Appointment Module version 1.
* Operating System: Windows, Linux
* Browser: chrome, firefox
* Test data: Sample resources with various availability and service combinations.

### **2. Test Strategy**

The testing will follow a combination of manual and automated testing approaches. Manual testing will be performed for functional testing, while automation will be used for repetitive tasks and regression testing.

* **API Testing:** Test all the developed APIs even before they are integrated with the UI.
* **UX Testing (Usability/User Experience Testing):** Test the user-friendliness of the application.
* **Automation Testing:** Run available automation scripts.
* **Smoke Testing:** Run smoke tests on each new build.

### **3. Test Cases**

### **4. Test Execution Schedule**

The following test execution schedule is proposed:

* Test Case 1: Add resource - [3/9/2023]
* Test Case 2: Edit resources - [4/9/2023]
* Test Case 3: View all resources in a list -[4/9/2023]
* Test Case 4: Delete resource - [5/9/2023]
* Test Case 5: Specify resource availability and services - [5/9/2023]
* Test Case 6: Add service- [7/9/2023]
* Test Case 7: Edit services- [8/9/2023]
* Test Case 8: View all services in a list - [9/9/2023]
* Test Case 9: Delete service- [9/9/2023]
* Test Case 10: Specify service availability and services - [10/9/2023]
* Test Case 11: Online appointment booking- [11/9/2023]
* Test Case 12: Add multiple appointments - [12/9/2023]
* Test Case 13: Add block time for a resource - [ 13/9/2023]
* Test Case 14: Reschedule existing appointments

during block time[13/9/2023]

* Test Case 15: View all existing time blocks - [15/9/2023]

### **5. Test Environment Setup/Teardown**

* Ensure that the Appointment Module is installed and accessible.
* Prepare the test data for resources with various availability and service combinations.

### **6. Defect Reporting**

Any defects encountered during testing should be reported using the standard defect reporting process. Defect reports should include detailed information on the issue, reproducing steps, and supporting evidence (if applicable).

### **7. Test Completion Criteria**

The following criteria must be met to consider the testing for resource management as complete:

* All test cases have been executed.
* All high and critical defects have been resolved or mitigated.
* The acceptance criteria have been met.