Test Plan - Restful Booker Application

**Created by:** Snehal Milind Tambe

## **1. Objective**

This document outlines the test plan for the hotel booking application. The objective is to ensure that all features and functionalities work as expected for the target audience.

## **2. Scope**

The scope of this test plan includes:

* **Features to be tested:**
  + Create Token
  + Get Booking ID(s)
  + Get Booking
  + Create Booking
  + Update Booking (Full Update)
  + Update Booking (Partial Update)
  + Delete Booking
  + Health Check
* **Types of testing:**
  + Manual Testing
  + Automated Testing (API automation)
  + Performance Testing
  + Accessibility Testing (where applicable)
* **Environments:**
  + Different browsers (for API testing tools like Postman/Swagger UI)
  + Operating systems (Windows, macOS, Linux)
* **Evaluation criteria:**
  + Number of defects found
  + Time taken to complete testing
  + API response time and reliability
  + User satisfaction ratings
* **Team roles and responsibilities:**
  + Test Lead: Oversees the testing process
  + Testers: Execute test cases and report defects
  + Developers: Fix defects and support testing
  + Project Manager: Manage timelines and resources

## **3. Inclusions**

* **Introduction:** Overview of the test plan, including purpose, scope, and goals.
* **Test Objectives:** Specific objectives such as identifying defects, ensuring API reliability, maintaining security standards, and achieving performance benchmarks.

## **4. Exclusions**

* UI-level testing of web pages (since primary focus is on API endpoints).
* Non-documented API endpoints not mentioned in the provided documentation.
* Load testing beyond agreed thresholds.

## **5. Test Environments**

* **Operating Systems:** Windows 10/11, macOS Ventura/Sonoma, Ubuntu Linux 22.04
* **Browsers:** Google Chrome, Mozilla Firefox, Microsoft Edge (for API tools access)
* **Devices:** Desktop computers, laptops
* **Network Connectivity:** High-speed Wi-Fi, wired Ethernet
* **Hardware/Software Requirements:**
  + Minimum 8 GB RAM
  + API testing tools like Postman, JMeter
  + Python or Java with REST Assured for automated API tests
* **Security Protocols:** Use of bearer tokens for authentication
* **Access Permissions:** Limited access with role-based permissions for testers and developers

## **6. Defect Reporting Procedure**

* **Criteria for identifying defects:**
  + API returns wrong data
  + Incorrect status codes (e.g., 500 error where 400 expected)
  + Authentication issues
  + Performance thresholds not met
* **Steps for reporting defects:**
  + Record steps to reproduce
  + Provide screenshots and API request/response payloads
  + Assign appropriate severity and priority
* **Triage and prioritization:**
  + Critical (blocking testing)
  + High (major feature impacted)
  + Medium (workaround available)
  + Low (minor impact)
* **Tracking tools:**
  + JIRA for issue tracking
* **Roles and responsibilities:**
  + Testers log bugs
  + Developers analyze and resolve bugs
  + Test Lead monitors progress
* **Communication channels:**
  + Email updates
  + Daily Standups
* **Metrics:**
  + Defect Density
  + Mean Time to Repair (MTTR)
  + Test Coverage %

## **7. Test Strategy**

**Step 1: Test scenarios and test cases creation:**

* Techniques:  
  + Equivalence Class Partitioning
  + Boundary Value Analysis
  + State Transition Testing
  + Use Case Testing
* Additional methods:  
  + Error Guessing
  + Exploratory Testing

**Step 2: Testing procedure:**

* Smoke Testing: Confirm critical endpoints are reachable.
* In-depth Testing: Execute detailed test cases for all endpoints.
* Multiple environments: Test APIs on different network settings (home, office Wi-Fi).
* Defect Reporting: Daily defect logging and updates.

**Types of Testing:**

* Smoke Testing
* Sanity Testing
* Regression Testing
* Retesting
* Functionality Testing
* Performance Testing (API response time validation)

**Step 3: Best Practices:**

* Context Driven Testing
* Shift Left Testing
* Exploratory Testing for hidden scenarios
* End-to-End API Flow Testing (from booking creation to deletion)

## **8. Test Schedule**

| **Task** | **Start Date** | **End Date** |
| --- | --- | --- |
| Test Plan Preparation | April 28, 2025 | April 29, 2025 |
| Test Case Creation | April 30, 2025 | May 1, 2025 |
| Test Execution (Manual) | May 2, 2025 | May 6, 2025 |
| Test Automation Setup | May 2, 2025 | May 7, 2025 |
| Test Report Submission | May 8, 2025 | May 9, 2025 |

## **9. Test Deliverables**

* Test Plan Document
* Test Scenarios and Test Cases
* Bug/Defect Reports
* Daily/Weekly Status Reports
* Test Summary Report

## **10. Entry and Exit Criteria**

**Requirement Analysis:**

* **Entry:** Requirement document shared and reviewed.
* **Exit:** All ambiguities clarified.

**Test Execution:**

* **Entry:** Test Cases created and signed off; Stable Build ready.
* **Exit:** All critical bugs fixed and retested.

**Test Closure:**

* **Entry:** All planned tests executed.
* **Exit:** Final Test Summary Report signed off.

## **11. Tools**

* JIRA Bug Tracking Tool
* Postman (Manual API Testing)
* Newman (Postman CLI Runner)
* JMeter (Performance Testing)
* Word and Excel (Documentation)
* Mind Map Tool for Test Scenario Mapping

## **12. Risks and Mitigations**

| **Risk** | **Mitigation** |
| --- | --- |
| Non-availability of Build URL | Have backup builds and plan buffer time. |
| API Downtime | Retry mechanism in automated scripts, Inform stakeholders immediately. |
| Resource Availability | Have a secondary resource trained as a backup. |
| Limited Testing Time | Prioritize critical endpoints first. |

## **13. Approvals**

Documents to be submitted for Client Approval:

* Test Plan
* Test Scenarios
* Test Cases
* Test Reports