Test Plan for Hotel Booking API

Created by: Biju Unnikrishnan

---

1. Objective

This document outlines the test plan for the **Hotel Booking API** application.
The objective is to ensure that all API endpoints for **hotel booking creation, retrieval, update, and deletion** work correctly for the target audience — **mobile apps, web apps, and third-party partners** using this API.

---

2. Scope

- **Features to be tested**:
  - Create Booking (POST `/booking`)
  - Retrieve Booking by ID (GET `/booking/{id}`)
  - Retrieve All Bookings (GET `/booking`)
  - Update Booking (PUT `/booking/{id}`)
  - Partial Update Booking (PATCH `/booking/{id}`)
  - Delete Booking (DELETE `/booking/{id}`)
  - User Authentication (POST `/auth`)
- **Types of Testing**:
  - Manual API Testing
  - Automated API Testing

    - Performance Testing (Load/Stress)
    - Security Testing (Authentication, Authorization)
- **Environments**:
    - Local and Staging Servers
    - Cloud Environments if applicable
- **Evaluation Criteria**:
    - API Response times
    - Accuracy and Integrity of data
    - Adherence to API standards and specifications
- **Team Roles**:
    - Test Lead, Manual Testers, Automation Testers, Developers

---

3. Inclusions

- Testing Positive and Negative scenarios
- Full CRUD operations for Booking APIs
- Token-based authentication handling
- Schema validation for request and response payloads

---

4. Exclusions

- UI Testing (if the API is backend-only)
- Payment Gateway Integration (if handled separately)
- Load Testing beyond 10,000 concurrent users

---

5. Test Environments

- **Operating Systems**: Windows 10, macOS
- **Tools**: Postman, Java with RestAssured, JMeter
- **Browsers**: Chrome/Firefox (only if accessing API documentation)
- **Devices**: Desktop and Laptop
- **Network**: Wi-Fi, 4G
- **Security**: Basic Authentication / Bearer Token
- **Access Permissions**: API keys and Role-based access if applicable

---

6. Defect Reporting Procedure

- **Criteria**: Incorrect API response, missing fields, authentication failure, status code mismatch
- **Reporting**: Using JIRA with proper steps and API logs
- **Triage**: Based on API importance (Critical, Major, Minor)
- **Tracking Tools**: JIRA, Bugzilla
- **Roles**: Testers report, Developers fix, Lead verifies

- **Communication**: Email updates or daily stand-up updates
- **Metrics**: Number of defects, average resolution time, reopening rate

---

7. Test Strategy

- **Step 1: Test Scenario and Test Case Creation**:
  - Techniques: Equivalence Partitioning, Boundary Value Analysis, Use Case Testing
- **Step 2: Testing Procedure**:
  - Smoke Testing: Validate Authentication and basic Booking Creation
  - Functional Testing: Validate all CRUD operations
  - Negative Testing: Missing fields, invalid data, unauthorized access
  - Regression Testing: After each bug fix
  - Performance Testing: Check API under load
- **Best Practices**:
  - Use dynamic data wherever possible
  - Validate both headers and body responses
  - Consistently monitor authentication validity

---

8. Test Schedule

Task	Duration
Test Planning	1 Day

Test Case Creation	2 Days
Automation Scripting	3 Days
Execution (Manual & Automation)	4 Days
Defect Retesting and Closure	2 Days

---

9. Test Deliverables

- Approved Test Plan
- Detailed Test Cases
- Automated Test Scripts
- Defect Reports
- Final Test Summary Report

---

10. Entry and Exit Criteria

- **Requirement Analysis**
  - Entry: Availability of API Documentation
  - Exit: Well-documented test scenarios and cases
- **Test Execution**
  - Entry: Stable APIs available
  - Exit: All critical test cases executed
- **Test Closure**
  - Entry: All defects logged and addressed

    - Exit: Test Summary Report delivered

---

11. Tools

- Postman (Manual API Testing)
- RestAssured with Java (Automation Testing)
- JIRA (Bug Tracking)
- JMeter (Performance Testing)
- Git (for version control)

---

12. Risks and Mitigations

Risk	Mitigation
API instability during early sprints	Close collaboration with developers
Token Expiry during tests	Implement auto-refresh in automation
Slow API Response times	Early performance testing

---

13. Approvals

- Test Plan Approval
- Test Scenario and Test Case Approval
- Final Test Summary Approval by QA Manager