Test Plan for IndiGo Mobile Application - Flight Booking

1. Introduction

- **1.1 Purpose:**
  - The purpose of this document is to outline the test plan for the IndiGo mobile application, specifically focusing on the flight booking functionality. This plan will ensure that the application meets the required functional, performance, and usability standards before release.
- **1.2 Scope:**
  - This test plan covers the following areas of the IndiGo mobile application:
    - Flight search
    - Flight selection
    - Passenger details entry
    - Seat selection
    - Ancillary services (meals, baggage, insurance)
    - Payment gateway integration
    - Booking confirmation
    - Mobile app installation and login/signup
- **1.3 Out of Scope:**
  - This test plan does not cover:
    - Flight check-in
    - Flight status updates
    - Loyalty program integration (IndiGo Blu)
    - Customer support features
    - App performance on tablets.
    - Specific device models will be selected for testing

2. Test Strategy

- **2.1 Methodology:**
  - A combination of manual and automated testing will be employed.
  - **Manual Testing:** Will be used for exploratory testing, usability testing, and functional testing of complex scenarios.
  - **Automation Testing:** Will be used for regression testing, performance testing, and repetitive functional tests. Tools like Appium will be considered.
- **2.2 Types of Testing:**
  - **Functional Testing:** Verify that all flight booking functions work as expected.
  - **Usability Testing:** Evaluate the app's ease of use and user experience.
  - **Compatibility Testing:** Ensure the app works correctly on different mobile devices (iOS and Android) and OS versions.

  - **Performance Testing:** Assess the app's responsiveness, load times, and stability under normal and peak conditions.
  - **Security Testing:** Identify potential security vulnerabilities, especially in payment processing and data handling.
  - **Regression Testing:** Ensure that new changes do not negatively impact existing functionality.
  - **Integration Testing:** Verify the correct interaction between different modules, especially with the payment gateway and the IndiGo backend systems.
  - **Accessibility Testing:** Verify the app adheres to accessibility guidelines.
- **2.3 Entry Criteria:**
  - The build to be tested must be stable with all planned features implemented.
  - Test environment (devices, network, test data) should be ready.
  - Test cases for the specified scope should be documented and reviewed.
- **2.4 Exit Criteria:**
  - All planned tests have been executed.
  - All critical and high-priority defects are resolved and verified.
  - The application meets the defined acceptance criteria.
  - A test summary report has been approved.

3. Test Environment

- **3.1 Hardware:**
  - A variety of mobile devices (smartphones) covering different manufacturers, screen sizes, and OS versions (both iOS and Android). Specific models will be defined in the detailed test plan.
- **3.2 Software:**
  - Operating Systems: Specific versions of iOS and Android as per the compatibility matrix.
  - Browsers: If any web views are used within the app.
  - Testing Tools:
    - Appium (for automation)
    - JMeter/LoadRunner (for performance testing)
    - ADB and Xcode tools (for debugging)
- **3.3 Network:**
  - Various network conditions, including Wi-Fi (different speeds), 4G, 3G, and simulated poor connectivity.
- **3.4 Test Data:**
  - Valid and invalid passenger data (names, ages, contact details).
  - Different flight routes, dates, and times.

  - Various payment methods (credit cards, debit cards, UPI, net banking).
  - Test data for ancillary services (meal preferences, baggage allowance).
  - Data for different user types (adults, children, infants, senior citizens).
  - Voucher codes and promotional offers.

4. Test Cases

- A detailed test case document will be created, covering the following scenarios (this is an example, and more detailed test cases will be added):

* **4.1 Flight Search**
   * Verify one-way, round trip and multi-city search functionality.
   * Verify search by city, airport code.
   * Verify date selection and calendar functionality.
   * Verify passenger count selection (adults, children, infants).
   * Verify search results display with correct flight details (time, duration, price).
   * Verify sorting and filtering of search results (price, departure time, arrival time, airline).
   * Verify error message for invalid search criteria.
   * Verify the functionality of the "View Fare Calendar" option.
   * Verify that the application remembers the last search.
* **4.2 Flight Selection**
   * Verify the ability to select a flight from the search results.
   * Verify fare details for the selected flight.
   * Verify seat availability.
   * Verify fare type selection (e.g., Saver, Flexi).
   * Verify the display of baggage allowance for the selected fare.
   * Verify the ability to navigate back to the search results.
* **4.3 Passenger Details Entry**
   * Verify the input fields for passenger information (name, date of birth, gender, contact details).
   * Verify the validation of input fields (e.g., email format, phone number format).
   * Verify the ability to add multiple passengers.
   * Verify the ability to save passenger details for future bookings.
   * Verify the functionality for entering contact details.
   * Verify the ability to select special assistance if needed.
* **4.4 Seat Selection**
   * Verify the display of the seat map.
   * Verify the ability to select available seats.
   * Verify seat selection charges.
   * Verify the ability to change selected seats.

   * Verify seat availability updates in real-time.
   * Verify the ability to skip seat selection.
* **4.5 Ancillary Services**
   * Verify the ability to add meals, baggage allowance, and travel insurance.
   * Verify the display of prices for ancillary services.
   * Verify the ability to select meal preferences.
   * Verify baggage allowance selection and charges.
   * Verify the ability to opt-in/opt-out of travel insurance.
   * Verify the ability to view details of the selected ancillary services.
* **4.6 Payment Gateway Integration**
   * Verify the integration with different payment gateways (credit cards, debit cards, UPI, net banking, wallets).
   * Verify the ability to enter payment details securely.
   * Verify the processing of successful and failed payments.
   * Verify the handling of payment errors (e.g., insufficient funds, invalid card details).
   * Verify the ability to save card details for future use (if applicable and with user consent).
   * Verify the application's behavior during network interruptions during payment.
   * Verify that the correct amount is charged.
* **4.7 Booking Confirmation**
   * Verify the display of the booking confirmation details (PNR, flight details, passenger details, payment details).
   * Verify the ability to view the booking confirmation.
   * Verify the option to receive the booking confirmation via email/SMS.
   * Verify the ability to download the ticket.
   * Verify the ability to navigate to the "My Bookings" section.
* **4.8 Mobile App Installation and Login/Signup**
   * Verify the app installation process from the respective app store (iOS App Store, Google Play Store).
   * Verify the app launch and splash screen.
   * Verify the signup process (email, mobile number) and account creation.
   * Verify the login process with valid and invalid credentials.
   * Verify the "Forgot Password" functionality.
   * Verify session management (e.g., session timeout).
   * Verify social login integration (if applicable).
   * Verify the ability to update profile information.
   * Verify that the app asks for necessary permissions.

5. Test Deliverables

- Test Plan document
- Test Case Specification document
- Test Execution Report
- Defect Report
- Test Summary Report

6. Roles and Responsibilities

- **Test Manager:** Responsible for overall test planning, execution, and reporting.
- **Test Lead:** Responsible for test case design, execution, and defect management.
- **Test Engineers:** Responsible for executing test cases, identifying defects, and documenting test results.
- **Developers:** Responsible for fixing defects and providing builds for testing.
- **Business Analyst:** Provides requirements and clarifies functionality.

7. Test Schedule

- A detailed test schedule will be created, outlining the timelines for test planning, test case development, test execution, and defect resolution. This schedule will be dynamic and will be updated as the project progresses.

8. Defect Management

- A defect tracking system will be used to log, track, and manage defects.
- Defects will be prioritized based on their severity and impact.
- A defect lifecycle will be followed (e.g., New, Assigned, Fixed, Verified, Closed).

9. Risk and Mitigation

- **Risk:** Tight deadlines may impact thorough testing.
    - **Mitigation:** Prioritize critical test cases, automate where possible, and allocate sufficient resources.
- **Risk:** Incompatibility issues with certain devices or OS versions.
    - **Mitigation:** Ensure a comprehensive device lab is available and perform compatibility testing early in the cycle.
- **Risk:** Issues with the test environment (e.g., network connectivity, server downtime).
    - **Mitigation:** Plan for environment setup and maintenance, and have contingency plans in place.
- **Risk:** Changes in the Application requirements.
    - **Mitigation:** Follow a Change Management process.

10. Approval

- The test plan will be approved by the Test Manager, Project Manager, and other stakeholders before test execution begins.