# TEST PLAN

| SECTION | DETAILS |
|---|---|
| Project Title | Airline Application |
| Prepared By | Vidyashree C V |
| Date | 05/05/2025 |

## 1. INTRODUCTION

The airline application facilitates flight bookings, reservation management, flight status tracking, and loyalty program interactions. This test plan outlines a structured validation approach, incorporating both manual and automation testing to ensure application reliability.

### 1.1 PURPOSE

To ensure the airline application is functional, secure, scalable and provides an optimal user experience, enabling passengers to book and manage flights seamlessly.

### 1.2 OBJECTIVES

- Validate core functionalities, including flight booking, modification, scheduling and cancellation.
- Ensure security compliance for user authentication and payment processing.
- Assess application performance under peak booking conditions.
- Verify cross-browser and cross-device compatibility (web & mobile).
- Identify and resolve defects before production deployment.

## 2. SCOPE

### 2.1 IN-SCOPE FEATURES

- Flight search and booking functionality.
- Payment processing and refunds.
- User account management and authentication.
- Flight status updates and notifications
- Loyalty program features.
- Mobile and web versions.

### 2.2 OUT-OF-SCOPE FEATURES

- Internal airline crew management.
- Third-party integrations (e.g., hotel or car rental bookings).
- Backend airline scheduling operations.

## 3. TESTING STRATEGY

### 3.1 MANUAL TESTING APPROACH

Manual testing will be employed for scenarios requiring human judgment and exploratory analysis:

- **Exploratory Testing**: Identifying defects without predefined test cases.
- **Usability Testing**: Evaluating user experience and intuitive navigation.
- **Ad-hoc Testing:** Quick checks on newly deployed features.

- **Regression Testing:** Ensuring existing functionalities remain intact after updates.
- **Cross-Browser Testing:** Validating application behavior across multiple browsers.

### 3.2 AUTOMATION TESTING APPROACH

Automation will be implemented for repetitive and critical validations:

- **Smoke Testing:** Quick verification of major functionalities post-build.
- **Functional Testing:** Automating workflows such as flight booking and payments.
- **Regression Testing**: Running automation scripts post-update to maintain system integrity.
- **Performance Testing:** Simulating high traffic conditions using JMeter or LoadRunner.
- **Security Testing:** Automating vulnerability analysis for payment transactions and authentication.

### 3.3 LEVELS OF TESTING

- **Unit Testing:** Validate individual components (e.g., flight search, booking, payments).
- **Integration Testing:** Verify module interactions (e.g., booking, payments, notifications).
- **System Testing:** Test entire application workflows end-to-end.
- **Performance Testing:** Evaluate system behavior under high user load.
- **Security Testing:** Ensure encryption, secure login, and payment compliance.
- **User Acceptance Testing (UAT):** Validate usability from a real customer perspective
.

## 4. EXECUTION STRATEGY

- Use agile methodology to allow iterative testing and continuous feedback.
- Automate repetitive test cases using selenium, Appium or Jmeter.
- Perform Manual Exploratory Testing to evaluate user experience.
- Implement negative testing to verify system handling of invalid inputs.
- Log defects, track resolutions and conduct post-fix regression tests.

## 5. ENVIRONMENT REQUIREMENTS

- Web and mobile versions (iOS, Android).
- Supported browsers (Chrome, Edge, Firefox, Safari).
- Database servers for transaction storage.
- Internet connectivity for real-world testing scenarios.

## 6. TEST ENVIRONMENT

- Hardware: Servers, databases, client devices
- Software: Airline Application, API services and automation tools.
- Networks: Secure environment with encrypted communication and firewalls.

## 7. ENTRY AND EXIT CRITERIA

**Entry Criteria:**

- Application build is ready for testing.
- Test data and environment are set up.
- Requirement documents and test cases are reviewed.

**Exit Criteria:**

- All critical defects are resolved.
- Functional, security and performance benchmarks are met.

- Final test reports and sign-off are completed.

## 8. TEST DELIVERABLES

- Test cases, Test scripts and Test data.
- Defect reports and resolution logs.
- Final Test summary report.

## 9. TEST SCENARIOS

- **Flight Booking & Management**
  1. Verify flight search functionality based on location and date.
  2. Validate flight booking across multiple payment options.
  3. Ensure confirmation emails and notifications are generated correctly.
  4. Test cancellation and refund processes.
  5. Verify flight modifications, including date and seat selection changes
- **Payment Processing**
  1. Validate successful and failed transactions with appropriate error handling.
  2. Ensure payment security and data encryption compliance.
  3. Test refund processing timelines and accuracy.
- **User Authentication & Account Management**
  1. Verify new user registration and email validation.
  2. Validate login/logout functionality with correct and incorrect credentials.
  3. Test Password reset and multi-factor authentication flows.
- **Flight Status & Notifications**
  1. Verify real-time flight status updates and delay notifications.
  2. Check push notifications and email alerts for flight changes.
- **Performance & Security**
  1. Simulate high traffic scenarios during peak hours.
  2. Conduct security tests for SQL injection, cross-site scripting (XSS) and authentication vulnerabilities.

## 10. RISK & MITIGATION

- Risk: High traffic during sales events may slow response times.
- Mitigation: Conduct extensive load testing before deployment.
- Risk: Security vulnerabilities in payment processing.
- Mitigation: Perform penetration testing and compliance checks.
- Risk: Integration failures with third-party services.
- Mitigation: Validate API response accuracy and fallback mechanisms.

## 11. DEPENDENCIES

- Access to airline flight schedules and database.
- API integrations with payment gateways.
- Authentication and session management systems.