**Test Plan for Airline Management System**

**1. Introduction**

This test plan outlines the strategy and scope for testing the Airline Management System (AMS), ensuring the software meets functional and non-functional requirements, and performs reliably under different conditions.

**2. Objectives**

* Verify all features of AMS function correctly.
* Identify and resolve system bugs or issues.
* Validate performance, security, and usability.
* Ensure regulatory compliance (e.g., aviation rules, data privacy).

**3. Scope**

**In Scope**

* Flight scheduling and management
* Booking and ticketing
* Check-in and boarding process
* Payment processing
* Admin and reporting modules
* User and access management

**Out of Scope**

* Third-party systems not directly integrated (e.g., external CRM)
* In-flight services

**4. Testing Strategy**

**4.1 Types of Testing**

* **Functional Testing** (UI, APIs, Business logic)
* **Integration Testing**
* **System Testing**
* **Regression Testing**
* **Performance Testing**
* **Security Testing**
* **User Acceptance Testing (UAT)**

**4.2 Test Levels**

* **Unit Tests**: Developers to test individual modules.
* **Integration Tests**: Interaction between modules (e.g., booking & payment).
* **System Tests**: End-to-end workflows.
* **Acceptance Tests**: Real-world user scenarios.

**5. Test Environment**

* **Hardware**: Simulated airline server setup, client devices
* **Software**: Latest build of AMS, browser versions, API tools (Postman, Swagger)
* **Database**: Test DB with sample passenger, flight, and ticket data

**6. Test Deliverables**

* Test Plan Document
* Test Cases & Scripts
* Bug Reports
* Test Summary Report
* Traceability Matrix

**7. Entry & Exit Criteria**

**Entry Criteria**

* All development completed
* Test environment configured
* Test data prepared

**Exit Criteria**

* All critical defects resolved
* 95%+ test case pass rate
* Test summary approved by stakeholders

**8. Responsibilities**

* **Test Lead**: Coordinates test activities, reporting
* **Testers**: Create and execute test cases
* **Developers**: Fix reported defects
* **Business Analysts**: Validate requirements

**9. Risks & Mitigation**

| **Risk** | **Mitigation** |
| --- | --- |
| Delay in development | Agile testing in parallel |
| Test data unavailability | Use mock or anonymized data |
| Integration failures | Early API contract testing |

**10. Schedule**

| **Phase** | **Start Date** | **End Date** |
| --- | --- | --- |
| Test Planning | May 1 | May 5 |
| Test Case Design | May 6 | May 12 |
| Execution | May 13 | May 25 |
| Reporting & Closure | May 26 | May 30 |

Would you like a detailed list of test cases or sample UAT scenarios next?