Requirement ID	Requirement Description	Design Reference	Development Reference	Testing Reference
FUNCTIONAL REQUIREMENTS				
REQ-001	The application shall allow users to input airport codes and layovers.	GUI Design Document, Input Form Design	MainCode.java, FlightInputController.java	Unit Test: InputValidationTest.java
REQ-002	The application shall display a list of airlines based on user input.	Flight Search Module Design	FlightSearch.java, AirlineListController.java	Integration Test: AirlineDisplayTest.java
REQ-003	The application shall calculate the most cost-effective and shortest flight routes.	Route Optimization Algorithm Design	RouteOptimizer.java, CostCalculator.java	System Test: RouteOptimizationTest.java
REQ-004	The application shall prompt users for their budget and check if the selected airline meets this budget.	Budget Handling Design	BudgetManager.java, FlightSelectionController.java	Unit Test: BudgetValidationTest.java
REQ-005	The application shall provide a visual representation of flight details and animations.	Visualization and Animation Design	FlightVisualizer.java, AnimationController.java	UI Test: FlightVisualizationTest.java
REQ-006	The application shall allow users to input and manage arrival and departure times.	Time Input Management Design	TimeInputManager.java, TimeController.java	Integration Test: TimeManagementTest.java
REQ-007	The application shall be developed using JavaFX for the graphical user interface.	JavaFX Implementation Guide	MainApp.java, GUIComponents.java	System Test: GUITest.java
REQ-008	The application shall interface with external APIs or databases for flight and airline information.	API Integration Design	APIConnector.java, DataFetcher.java	Integration Test: APIIntegrationTest.java
REQ-009	The application shall handle errors gracefully and provide user feedback.	Error Handling Design	ErrorHandler.java, FeedbackManager.java	Unit Test: ErrorHandlingTest.java
REQ-010	The application shall be compatible with major operating systems (Windows, macOS, Linux).	Compatibility Design	PlatformCompatibilityChecker.java	System Test: OSCompatibilityTest.java
NON FUNCTIONAL REQUIREMENTS				
NFR-01	The system should handle route optimization within 5 seconds	Route Optimization Module	Test Case	Performance test for time taken to calculate routes
NFR-02	The system must support multiple user requests simultaneously	User Input Handling, Backend Controller	Test Case	Load testing to assess concurrent user handling
NFR-03	Ensure secure handling of user data	Security Module, Data Management	Test Case	Validate data encryption, and secure data access
NFR-04	Protect against unauthorized access and data breaches	Authentication & Authorization Module	Test Case	Test for secure login, roles, and permission checks
NFR-05	Provide a user-friendly interface with clear instructions	User Interface (JavaFX)	Test Case	Usability testing, user feedback, and accessibility checks
NFR-06	The application should be intuitive and require minimal training	User Interface (JavaFX)	Test Case	Conduct user acceptance testing (UAT)
NFR-07	System should scale to handle more users and data	Database, Backend Services	Test Case	Scalability test by simulating higher loads
NFR-08	Include mechanisms for recovery in case of failure	Error Handling and Recovery Module	Test Case	Simulate failures and check for recovery processes
NFR-09	System should maintain minimal downtime	Deployment and Infrastructure Management	Test Case	Monitor uptime using automated tools
NFR-10	Consistent performance with accurate results	Route Optimization Module, Flight Visualization	Test Case	Perform regression and reliability testing