Scope: The test plan focuses on validating the accuracy of the flight cost calculation system. It includes testing the calculation of costs based on distance, departure date, and service class. It also includes testing additional costs for extra baggage.

Objectives:

1. Verify that the flight cost is calculated accurately based on the given criteria.
2. Ensure that costs are calculated correctly for different flight distances, departure dates, and service classes.
3. Validate the calculation of additional costs for extra baggage.
4. Identify and report any discrepancies or calculation errors in the system.

Test Techniques:

1. Equivalence Partitioning: Test the system with different input values representing different equivalence classes.
2. Boundary Value Analysis: Test the system with inputs at the boundaries of valid ranges to verify correct handling of edge cases.
3. Positive and Negative Testing: Validate the system with valid and invalid inputs to ensure proper error handling.
4. Regression Testing: Repeat tests on previous defects to ensure they have been resolved and new changes haven't introduced regressions.

Tools:

1. Test management tool Jira, TestRail for test case management and tracking.
2. Test automation framework (Selenium) for automating repetitive tests.
3. Spreadsheet software Microsoft Excel for test data management and calculations.

Estimation: The test effort is estimated to be 40 hours, including test design, execution, and reporting.

Resources:

1. Testers: 2
2. Test environment with access to the flight cost calculation system.
3. Test data and test scenarios.
4. Test management tool for documenting and tracking test results.

Deliverables:

1. Test plan document.
2. Test cases with inputs, expected results, and actual results.
3. Test execution reports.
4. Defect reports for any identified issues.
5. Test summary report.

Risk and Assumptions:

1. Assumption: The flight cost calculation system accurately implements the given criteria.
2. Risk: Inaccurate flight cost calculation may lead to incorrect pricing for customers, resulting in financial loss and customer dissatisfaction.
3. Risk: Integration issues with external systems providing flight data may affect the accuracy of the calculated costs.
4. Risk: The system may have performance issues when processing a large volume of flight data.

Test Plan Execution:

1. Test Setup: a. Set up the test environment with the flight cost calculation system. b. Prepare test data and test scenarios for various distance ranges, departure dates, and service classes.
2. Test Case Design: a. Identify different equivalence classes for distance, departure date, and service class inputs. b. Create test cases based on equivalence partitioning, boundary value analysis, and positive/negative testing techniques. c. Include test cases for additional costs related to extra baggage.
3. Test Execution: a. Execute the test cases using both manual and automated approaches. b. Input the test data into the flight cost calculation system. c. Compare the calculated cost with the expected cost based on the given criteria. d. Record the actual results and any deviations from the expected results. e. Report any defects found during testing, including steps to reproduce and relevant screenshots or logs.
4. Regression Testing: a. Retest previously reported defects after they have been fixed. b. Repeat relevant tests from the initial test cycle to ensure new changes haven't introduced regressions.
5. Test Reporting: a. Document the test results, including actual results, deviations, and any issues found. b. Create a test summary report summarizing the test execution, overall results, and any recommendations for improvement.
6. Test Closure: a. Review the test results and obtain necessary approvals. b. Provide the test summary report to stakeholders. c. Archive the test artifacts for future reference.