Test Plan

1. **Introduction**
   1. Test Plan Objectives

**a. Test Plan Objectives**

* + 1. **The project being tested is a comprehensive logistics management system designed to optimize the routing and scheduling of delivery trucks. The system, referred to as the "Logistics Routing System," integrates several key features including route planning, real-time tracking, and automated dispatching to improve the efficiency of delivery operations.**
    2. **Project Overview:**

**The Logistics Routing System is intended to streamline the process of planning and managing delivery routes for a fleet of trucks. The system will support multiple routes, handle various types of cargo, and ensure deliveries are made in a timely and efficient manner. It features:**

**A 2D map representation of the delivery area.  
Route optimization algorithms to determine the most efficient paths.  
Real-time updates and tracking of truck locations.  
Support for different types of cargo and truck capacities.**

* + 1. **Objectives for the Test:**

**The primary objective of the testing phase is to ensure the Logistics Routing System operates reliably and meets all specified requirements. The testing will be conducted at various levels, including system, performance, security, and user acceptance testing, to validate the system's functionality, performance, and security aspects.**

* + 1. **Testing Goals:**

**Functionality Testing: Verify that all features of the system, such as route planning, tracking, and dispatching, work as intended.**

**Performance Testing: Assess the system's performance under different load conditions to ensure it can handle the expected volume of deliveries without degradation in performance.**

**Security Testing: Ensure the system is secure from unauthorized access and vulnerabilities.**

**User Acceptance Testing: Confirm that the system meets the end-users' needs and requirements, and is user-friendly..**

1. **Scope**

**a. Items to be Tested**

**The following components and functionalities of the Logistics Routing System will be tested:**

1. **Map Representation and Route Planning:**
   1. **Accurate representation of the 2D map.**
   2. **Correct implementation of route planning algorithms.**
   3. **Display and update of routes on the map.**
2. **Real-Time Tracking:**
   1. **Real-time updates of truck locations on the map.**
   2. **Accuracy of tracking information.**
3. **Automated Dispatching:**
   1. **Correct assignment of deliveries to trucks based on capacity and route optimization.**
   2. **Proper handling of multiple deliveries and route adjustments.**
4. **System Performance:**
   1. **Load testing to ensure the system can handle multiple concurrent users and high volumes of data.**
   2. **Stress testing to evaluate system behavior under extreme conditions.**
5. **User Interface:**
   1. **Usability testing to ensure the interface is user-friendly and intuitive.**
   2. **Verification of all interactive elements and their functionality.**

**b. Items Not to be Tested**

**The following components and functionalities are outside the scope of this testing phase:**

1. **Third-Party Software: Any third-party tools or software integrated into the system will not be individually tested. The focus will be on the integration points and data exchange.**
2. **Hardware Performance: The performance and reliability of hardware (e.g., servers, network infrastructure) used to host the system will not be tested. Testing will assume the underlying hardware infrastructure is functioning correctly.**
3. **Test Strategy**

This section describes the approach we will take to perform the tests for the Logistics Management System. The testing process will be thorough and systematic, ensuring that all critical defects are identified and resolved before moving on to the next phase of testing

1. **Test Data**

Test data will be obtained from a combination of predefined datasets and dynamically generated data to ensure comprehensive coverage

**Levels of Testing**

1. **Exploratory Testing**

Objective: Identify and remove critical defects early in the testing process.

Approach: Testers will use their knowledge and experience to explore the system and identify major issues.

1. **Functional Testing**

Objective: Ensure all primary functions of the application are delivered correctly.

Approach: Execute predefined test cases that cover all functional requirements.

* 1. **System Test**

Objective: Validate the complete and integrated system to ensure it meets the specified requirements.

Approach: Test the system as a whole.

**3.2 Performance Test**

Objective: Verify the system's performance under various conditions.

Approach: Conduct load testing, stress testing.

**3.3 Security Test**

Objective: Identify vulnerabilities and ensure the system is secure against potential threats.

Approach: Perform penetration testing and security assessments.

**3.4 Automated Test**

Objective: Increase testing efficiency and coverage using automation tools.

Approach: Develop and execute automated test scripts for repetitive tasks.

**3.5 Stress and Volume Test**

Objective: Assess the system's stability and performance under extreme conditions.

Approach: Test high volumes of data and user activity to evaluate system behavior.

**3.6 Recovery Test**

Objective: Ensure the system can recover from failures and resume normal operations.

Approach: Test failure scenarios and validate the system's recovery mechanisms.

**3.7 Documentation Test**

Objective: Verify the accuracy and completeness of system documentation.

Approach: Review and test all user manuals, guides, and technical documents.

**3.8 Beta Test**

Objective: Gather feedback from end-users in a real-world environment.

Approach: Release a beta version to a select group of users and collect their feedback.

**3.9 User Acceptance Test (UAT)**

Objective: Confirm the system meets user requirements and expectations.

Approach: Conduct testing with end-users to validate functionality and usability.

1. **Test Design Process**
2. **Understanding Requirements**

Collaborate with stakeholders to gather and understand the system requirements.

Ensure all requirements are clear, complete, and testable.

1. **Build a Traceability Matrix**

Create a traceability matrix to map requirements to test cases.

Ensure every requirement is covered by one or more test cases.

1. **Prepare Test Cases**

Develop detailed test cases based on the requirements and traceability matrix.

Ensure test cases are comprehensive and cover all aspects of the system.

1. **Review Test Cases**

Have test cases reviewed by another member of the quality assurance team.

Ensure accuracy, completeness, and adherence to testing standards.

1. **Environment Requirements**
   1. **Hardware**: Development machines with sufficient RAM and processing power.
   2. **Software**: IDE (Integrated Development Environment), necessary libraries (e.g., for map visualization), and testing frameworks.
2. **Execution Strategy**
   1. this is the section where you will describe heavy chests are actually executed. You can describe what the entry and exit criteria for the tests are. For example you might be able to exit a test if it passes 95% of test scripts. In another situation, you might want to pass 100% of the tests. Or perhaps you want to declare but a test is completed if there are no severe or critical defects.
   2. You can describe the severity of defects in this section and break them down into severity levels of:

**Entry and Exit Criteria:** Start testing after code integration; exit after verifying core

functionalities without critical defects.

* 1. **Test Reporting**
     1. **Daily Test Reports:**  
        Contents: Number of tests conducted, passed, and failed. Brief descriptions of the areas tested and those that failed.  
        Frequency: Daily
     2. **Weekly Summary Reports:**  
        Contents: Summary of the week's testing activities, major defects found, and resolved issues. Analysis of test coverage and areas of concern.  
        Frequency: Weekly

1. **Test Schedule**

The following test schedule outlines the timeline for the testing activities of the Logistics Routing System (LRS) project. This schedule includes key milestones, testing phases, and estimated durations to ensure comprehensive testing is completed on time.

Milestones and Phases

* 1. Test Planning and Preparation
     + 1. Start Date: July 10, 2024
       2. End Date: July 15, 2024
       3. Activities:
          1. Finalize test plan
          2. Prepare test cases and scripts
          3. Set up test environment
          4. Review and approve test plan
  2. Unit Testing
     + 1. Start Date: July 16, 2024
       2. End Date: July 23, 2024
       3. Activities:
          1. Test individual modules/components
          2. Ensure each unit functions correctly in isolation
          3. Log and fix defects
  3. Integration Testing
     + 1. Start Date: July 24, 2024
       2. End Date: August 1, 2024
       3. Activities:
          1. Test integrated components and modules
          2. Ensure interfaces and interactions work correctly
          3. Log and fix integration defects
  4. System Testing
     + 1. Start Date: August 2, 2024
       2. End Date: August 15, 2024
       3. Activities:
          1. Perform end-to-end testing of the complete system
          2. Validate functional and non-functional requirements
          3. Log and fix system-level defects

1. **Control Procedures**

7.1 Reviews

* 1. Objective: Ensure accuracy and completeness of test plans, cases, and scripts.
  2. Process: Prepare documents, review in meetings, collect feedback, and update accordingly.
  3. Bug Review Meetings
  4. Objective: Review, prioritize, and assign reported bugs.
  5. Process: Prepare bug reports, discuss in meetings, assign fixes, and track progress.
  6. Change Request

Objective: Manage changes affecting testing.

Process: Submit, review, approve/reject changes, update documents, and communicate changes.

* 1. Defect Reporting
  2. Objective: Systematic defect reporting and resolution.
  3. Process: Detect, describe, submit, review, assign, track, and retest defects

1. **Functions To Be Tested**

**The functions that need to be tested are:**

**8.1 Capacity Calculation function:**

**The function's ability to correctly determine truck capacity will be tested under various cases, including different combinations of weight and box sizes, reaching maximum weight or volume, and managing trucks with different limitations. Test cases will make sure that the function can correctly calculate available capacity in all vulnerable situations.**

**8.2 Shipment Allocation function**

The function test will cover different situations including cases where there would be multiple trucks, or capacity limitation etc. These tests confirm that shipments are correctly assigned to trucks given their weight, dimensions, and destination without any discrepancies.

**8.3 Shortest Path Calculation Function**

The test cases for this function will make sure that it correctly gives the shortest path, while taking care of buildings. The vulnerable cases include the one in which starting or destination is at corners or the ones in which buildings block the way.

**8.4 Output Message Generation function**

The testing of this function makes sure that the messages prompted are clear with respect to truck selection, paths etc. Mainly the clarity and accuracy of messages would be tested.

1. **Deliverables**

**Resources and Responsibilities**  
 **9.1. Resources**

a) Testing Environment: A proper environment with all the required hardware, software and data.

b) Testers: People who will do the testing.   
**9.2. Responsibilities**

a) Testers: They will run test cases, record the results, compare them with expected results and generate report accordingly.

b) Developers: The report generated by testers will be used by developers to fix the required issues and resolve the bugs.

c) Project Managers: The main task for them is to oversee if testers and developers are coordinating properly.

1. **Deliverables**

a) Packages are correctly assigned to trucks considering distance, space and diversions and that to through the shortest path.

b) There must be minimal distance between the truck’s route and the destination.

c) Information printed regarding the truck assigned, destination for delivery and diversions should be clear and accurate.

d) Cases in which truck can’t reach destination should be addressed.

1. **Suspension / Exit Criteria**

a. Failure to correctly assign bundles to truck based on distance, space and diversions.

b. Failure to print data correctly.

c. Failure to allocate shortest path.

1. **Resumption Criteria**

**Once the developer team resolves the bugs, the program should be tested again if it gives required output. Also, take more cases into consideration as well because fixing some bugs might introduce some new ones.**

1. **Dependencies**  
   **12.1 Personnel Dependencies**

Requirement of testing team, developers, and end-users  
**12.2 Software Dependencies**

Requirement of necessary tools like automation tool, tracking tool and libraries for testing. Also includes the application to be tested.  
**12.3 Hardware Dependencies**

Equipment of the development and testing team must be working.  
**12.3 Test Data & Database**

Availability of valid and realistic test data is crucial. For this application, test data should cover different package sizes and weights, vulnerable destinations and other corner cases.

1. **Risks**  
   **13.1. Schedule**

**Delay in** resolving the bugs, which can be due to factors like test data availability, tight deadlines etc. This can be minimized by making proper schedule and maintaining proper communication.   
**13.2. Technical**

Technical risks include hardware and network issues, inadequate data or incompatible testing environment. These can be minimized by ensuring proper configuration of software and hardware.  
**13.3. Management**

Risk of lack of coordination between developers and testers. Communication is the key.  
**13.4. Personnel**

At times there can be conflict of opinions within the team members, which can bring down the motivation and harms the working environment. Ego clash is the main reason for such things. All members should respect other’s skill set and deal with patience.  
**13.5 Requirements**

Sometimes the requirements itself can be a challenging part. They may be not clear, or too difficult to understand quickly. Invest time and patience in such cases.

1. **Tools**

**For seamless testing, we require test management tools like Jira, test automation tools, performance testing tools and Security testing tools.**

1. **Documentation**

During testing we create documents like test plan (to provide an overview), test cases (data we are going to use), test scripts (to perform automation), defect logs (to record all bugs found) test reports (analyzing how much results match with expected outputs, giving us the quality of the software) and user manuals (mainly including troubleshooting tips).

1. **Approvals**

**Mainly, approvals are given by Project Manager, Development Lead and QA lead this ensures that all the departments involved in testing have reviewed the plan.**