**Test Plan**

**Introduction**

The purpose of this test plan is to outline the strategy and approach for testing the delivery route optimization program as part of Milestone 2. This milestone involves ensuring that the given software correctly assigns shipments to trucks based on their capacity and proximity to the delivery destination, using the A\* algorithm for route optimization.

**Test Plan Objectives**

* Validate the functionality of the provided code for assigning shipments to trucks.
* Ensure the A\* algorithm correctly calculates the shortest path for delivery trucks.
* Confirm the program correctly handles edge cases such as invalid inputs and full truck capacity.
* Identify and document any defects or areas for improvement.

**Scope**

**In Scope**

* Testing the assignment of shipments to trucks.
* Validating the calculation of the shortest path using the A\* algorithm.
* Ensuring correct handling of maximum truck capacity (weight and volume).
* Checking the program's response to various input scenarios.

**Out of Scope**

* Integration with other systems or external dependencies.
* Performance testing beyond basic functional validation.

**Test Strategy**

**System Test**

* Test the overall functionality of the program by simulating multiple shipments and ensuring correct truck assignments.

**Functional Test**

* Verify that shipments are correctly assigned to trucks based on capacity and route proximity.
* Check the correct implementation of the A\* algorithm for route calculation.

**Edge Case Test**

* Test the program's response to invalid inputs (e.g., invalid weight, size, or destination).
* Ensure proper handling when all trucks are at full capacity.

**Environment Requirements**

* Development and testing will be conducted on a local machine running a compatible IDE for C programming (e.g., Visual Studio).
* The test environment will include all necessary libraries and dependencies as specified in the provided code files.

**Execution Strategy**

**Entry Criteria**

* All required code files are present and compiled without errors.

**Exit Criteria**

* All test cases have been executed.
* All critical and high-severity defects have been resolved.

**Severity Levels**

* **Critical:** Causes system crash or incorrect shipment assignment.
* **High:** Incorrect route calculation or truck capacity handling.
* **Medium:** Incorrect handling of edge cases.
* **Low:** Minor UI issues or incorrect error messages.
* **Cosmetic:** Minor display issues that do not affect functionality.

**Test Reporting**

* Daily test reports will be generated and shared with the development team.
* The test report will include the number of tests executed, passed, failed, and a summary of defects.

**Test Schedule**

* **Day 1-2:** Review and understand the provided code.
* **Day 3-4:** Design test cases and set up the test environment.
* **Day 5-7:** Execute test cases and report findings.
* **Day 8:** Review and finalize the test plan and test cases.

**Control Procedures**

* Regular review meetings will be conducted to assess progress.
* Defect review meetings will be held to prioritize and address issues.

**Functions To Be Tested**

1. Assignment of shipments to trucks.
2. Calculation of the shortest path using the A\* algorithm.
3. Handling of truck capacity limits (weight and volume).
4. Response to invalid input scenarios.

**Resources and Responsibilities**

* **Test Lead:** Responsible for creating and maintaining the test plan and test cases.
* **Testers:** Execute test cases and report defects.
* **Developers:** Fix defects and provide feedback.

**Deliverables**

* Completed test plan.
* Test cases document.
* Daily test reports.
* Defect logs.

**Suspension / Exit Criteria**

* Testing will be suspended if a critical defect blocks further testing.
* Testing will exit once all planned test cases are executed, and critical/high defects are resolved.

**Resumption Criteria**

* Testing will resume once the blocking defect is resolved.

**Dependencies**

* **Personnel Dependencies:** Availability of the development and testing team.
* **Software Dependencies:** Required IDE and libraries for compiling and running the program.
* **Hardware Dependencies:** Access to a testing environment with necessary resources.

**Risks**

* **Schedule:** Delays in fixing critical defects.
* **Technical:** Unforeseen issues with the A\* algorithm implementation.
* **Management:** Coordination issues between development and testing teams.
* **Personnel:** Availability of key team members.

**Tools**

* IDE for C programming (e.g., Visual Studio).
* Version control system (e.g., GitHub).

**Documentation**

* Test plan document.
* Test cases document.
* Daily test reports.
* Defect logs.

**Approvals**

* Test plan approved by the project team leader and/or professor..