Test Plan

1. **Introduction** 
   1. Test Plan Objectives
      1. **The objective of this test plan is to make sure that the delivery algorithm precisely decides which truck the package will be placed in based on the weight and size of the box, print out where the truck will deliver the package and print the route that is the shortest and closest to the destination.**
2. **Scope** 
   1. **Assigning packages to trucks based on their weight and volume**
   2. **Finding which truck is the closest to the destination**
   3. **Calculating the shortest route of the point from the truck and the destination**
   4. **Directing situations when the truck cannot reach the destination**
   5. **Directing situations when the package is not delivered**
3. **Test Strategy** 
   1. **In our test strategy, we aim to focus on the functionality, performance, and the reliability of the delivery program. The goal is to ensure that the program meets all requirements, functions correctly under various conditions and is reliable at the same time. To focus on these main parts, system testing, performance testing, automated testing, documentation testing and user-acceptance testing will be conducted. Other tests like security, stress and volume testing, recovery testing and beta testing will not be conducted as it is not required on the project specifications. In addition, we plan to use unit testing or printing log statements to validate individual parts of the program. Our strategy focuses on the program's functionality meeting its requirements with the selected test cases.**

**Required Tests**

3.1. System Test – Ensures that all functions work as intended. It tests the integration to validate that different components of the system work together seamlessly as a whole.

3.2. Performance Test – Test the system’s ability to handle different situations where the package is not being able to be delivered or where the truck is full.

3.3. Automated Test – To save time and ensure consistency, automated repetitive tests like using log statements will be conducted for individual functions

3.4. Documentation Test – To make sure the documentation guides the testing processes and make sure it aligns with what the software is intended to do. This will ensure that the system’s features and performances are described clearly for the users.

3.5. User Acceptance Test – In order to validate the requirements and the expectations of the program, this test will be conducted to make sure that the program precisely aligns with the described scope of the program.

**Not Required Tests**

3.6. Security Test – Since the focus of the test is on functionality and performance, security tests will not be conducted.

3.7. Stress and Volume Test – Testing for the system’s load handling is not required much as performance testing covers most of it. So, this test will not be conducted.

3.8. Recovery Test – Testing for recovery on this system is not mentioned in this project, so this test will not be conducted.

3.9. Beta Test – This project focuses on internal validation first so beta testing may not be relevant.

* 1. **Test Design Process and Overview**

1. **Understand Requirements**

**The test design process ensures that all parts of the system are tested thoroughly. Understanding the requirements ensures that both functional and non-functional requirements are met and documented. It gives a comprehensive understanding of what needs to be tested the what the expected behaviors are.**

1. **Build a traceability matrix**

**A traceability matrix ensures that a clear link is provided between the requirements and the test cases. Therefore, building a traceability matrix will ensure that all requirements are covered by test cases which completes the test coverage.**

1. **Preparing test cases**

**Creating test cases for each specific requirement that are ready for execution allows for expected outcomes. Test cases are created that meets their requirements where they cover multiple scenarios from basic functionality to out-of-bound conditions to edge cases.**

1. **Test case review**

**Through peer review of the test cases, the test cases can be improved both in accuracy, quality, and effectiveness. Errors and inconsistencies can be identified through peer review, so the finalized test cases can be of higher quality and accurate for execution.**

1. **Environment Requirements** 
   1. Hardware Requirements:
      1. Hardware requirements that cover enough like sufficient RAM and storage to run the software are needed for Windows. Backup recovery is also required so that in case of machine failure, the code and test cases can be recovered.
   2. Software Requirements:
      1. The operating system for Windows should be the latest version.
      2. Visual Studio IDE will be used for debugging, writing, and maintaining the code.
      3. For version control, git integrated with GitHub will be used.

* 1. Test Harness:
     1. Pre-existing testing tools from official websites or open-source resources will be installed and used on test machines to conduct the tests.

1. **Execution Strategy** 
   1. Entry Criteria

The software build must be installed on test machines, with all hardware and software setups complete. The test environment should be ready, including available test data and reviewed, approved test designs (cases, scripts, documentation).

* 1. Exit Criteria

Testing concludes when all scripts pass without major defects, all test cases are executed and recorded, and stakeholders approve readiness for deployment or the next phase.

* 1. Severity Levels

1. **Critical**: System crashes or severe issues halt testing.
2. **High**: Major functionality loss with potential workarounds.
3. **Medium**: Quality issues with available workarounds.
4. **Low**: Minor functionality impact, such as unclear error messages.
5. **Cosmetic**: User interface issues with no functional impact.
   1. Test Reporting
6. **Reports**: Daily summaries of tests run, passed, and failed, including brief descriptions of tested and failing areas.
7. **Recipients**: Project manager, development team, QA team.
8. **Communication**: Testers report bugs to project managers, who assign fixes to developers. Regular updates via meetings, emails, and issue tracking systems.
9. **Test Schedule**
   1. **The tests can be broken down into smaller tests and can be tested based on those parameters, so it takes less time to complete. If we find a bug during testing the schedule can be adjusted accordingly. This will help us prioritize the more complex tasks so they can be completed in time. If anybody finds a bug, they will report it immediately so we can meet the deadlines in time.**
10. **Control Procedures**

6.1 Reviews- To guarantee the accuracy, completeness, and effectiveness of the testing process, it is imperative to conduct routine reviews of test cases and scripts.

6.2 Bug Review Meetings- regular bug review meetings are held to discuss and prioritize identified defects.  
6.3 Change Request- It is imperative to document and review change requests in order to effectively manage modifications to the project's scope, requirements, or functionality.

6.4 Defect Reporting- To document and monitor the resolution process, comprehensive defect reports are generated for each identified issue.

1. **Functions To Be Tested** 
   1. **Shipment Assigning Function:**
      1. **This function will assign packages based on their weight and volume correctly to trucks.**
   2. **Shortest Route Calculation Function:**
      1. **This function will calculate the shortest point between the point of the truck and the point of the destination all the while avoiding buildings.**
   3. **Truck Capacity Function**
      1. **This function will be used to test the available capacity of the truck for the packages.**
   4. **Output Message Function**
      1. **This function will be tested to make sure it generates the message for the current situation concerning packages, trucks and route.**
2. **Resources and Responsibilities**   
   8.1. Resources

One person from the group will get all the resources. Team members will make a list of all the resources required and that person can get them. The leader will make sure that no extra resources are asked for. This will make sure that we have all the resources on time.

8.2. Responsibilities

First When filling the scrum report all the responsibilities will be distributed by the group leader or team members can assign themselves the responsibilities. This will make sure that responsibilities are evenly distributed all over the group.

1. **Deliverables**

The deliverables for this test plan include:

1. Test Plan Document: This comprehensive document outlines the entire testing strategy, scope, objectives, and procedures, providing a roadmap for the testing process.
2. Test Cases: Detailed test cases for each functionality of the system, including expected outcomes, steps to execute, and criteria for pass/fail.
3. Test Scripts: Automated scripts for running repetitive and critical tests, ensuring consistency and saving time.
4. Test Data: All necessary data required to execute the test cases, including both valid and invalid data sets to thoroughly test the system's functionality and robustness.
5. Test Environment Setup: Configuration of hardware and software environments for testing, including the setup of necessary tools, frameworks, and platforms.
6. Test Execution Reports: Regular reports detailing the progress of the testing, including the number of test cases executed, passed, failed, and any defects found, along with their severity and status.
7. Defect Logs: Comprehensive logs of all defects found during testing, including severity, status, and resolution, providing a clear record for tracking and managing issues.
8. Final Test Report: A summary report at the end of the testing phase, including overall results, any remaining issues, recommendations for improvements, and final approval status from stakeholders.
9. **Suspension / Exit Criteria**

Suspension criteria define when testing activities may be paused, and exit criteria specify when testing is considered complete.

**Suspension Criteria:**

1. Critical Defects: When critical defects are identified that block further testing or significantly impact system functionality.
2. Hardware/Software Failures: Any failures in the hardware or software that prevent the continuation of tests.
3. Resource Unavailability: Unavailability of critical test environments, tools, or personnel needed for testing.
4. Environmental Issues: Situations such as power outages, network failures, or other environmental factors that disrupt testing.

**Exit Criteria:**

1. Completion of Test Cases: All planned test cases have been executed, and results have been recorded.
2. Defect Resolution: No critical or high-severity defects remain unresolved, and all medium and low-severity defects are documented and accepted by stakeholders.
3. Pass Rate: At least 95% of test cases have passed, ensuring a high level of confidence in the system's stability and functionality.
4. Documentation Completion: All test documentation, including test cases, test scripts, and defect logs, is complete and reviewed.
5. Stakeholder Approval: Final approval from stakeholders, indicating that the testing objectives have been met and the system is ready for deployment or further phases.
6. **Resumption Criteria**

Resumption criteria detail the conditions required to restart testing after it has been suspended.

1. Critical defects have been resolved, and fixes are verified.
2. Test environments are restored and fully functional.
3. Necessary resources are available and ready to continue testing.

These sections ensure that the testing process is well-documented, monitored, and controlled to meet the project's quality and timeline objectives.

1. **Dependencies**   
   12.1 Personnel Dependencies- The testing process is contingent upon the availability of critical team members, including developers, test engineers, and project managers. The testing schedule and the capacity to promptly resolve issues can be influenced by any unavailability due to illness, leave, or other commitments.

12.2 Software Dependencies- Various software tools, including automated testing tools, performance testing tools, and bug tracking tools (e.g., JIRA), are necessary for the testing process. Additionally, it is imperative to have access to the development, staging, and production environments in order to conduct a thorough testing process.  
12.3 Hardware Dependencies- Adequate hardware resources are required to conduct testing, which includes testing computers, servers, and network infrastructure. The accuracy of test results can be impacted by insufficient or malfunctioning hardware, which can cause delays in testing activities.

12.3 Test Data & Database - To verify the system's functionality and performance, it is imperative to have comprehensive and precise test data. Testing data storage, retrieval, and integrity necessitates database access. The testing process may be impeded by any issues with test data or database access.

1. **Risks**   
   13.1. Schedule -Delays in the development process could impact testing timelines. Delayed feedback or approval from stakeholders.

13.2. Technical - Integration issues between different modules., High user load results in performance bottlenecks.

13.3. Management - insufficient resources can impede the testing process. Adjustments to project requirements can lead to additional work and delays.

13.4. Personnel- Critical team members may be unavailable due to illness, leave, or other reasons.

13.5 Requirements- Constant changes can disrupt the testing process and lead to rework.

1. **Tools**

**These tools will be used for accurate testing and test management:**

1. **GitHub**
2. **Jira**
3. **Visual Studio**
4. **Documentation**

**During all these testing, everything can be documented through various documents like test cases, log files, scrum reports or output stored in files for reviewing.**

1. **Approvals**

**For approval, the test plan and the associated documents are reviewed by reviewers through review meetings and feedback on the test cases. Through this approval process and feedback, test cases will be done and updated.**