Test Plan

1. **Introduction**

This project is related to the delivery system that will be used by local delivery companies. We basically receive information from our customers, such as the weight and size of the shipment, and the destination of the delivery. Based on this information, the goal of this program is to find the truck with the best conditions and its route. Every truck has a fixed weight and volume. Each truck has its own truck with three different routes. Each truck can deliver through its designated route, or through space where no other truck's routes or buildings exist. Based on these conditions, the shortest possible distance for a truck is calculated. Compare the calculated distances to find the best route for the truck. If the distances are the same, we will find a truck that currently carries fewer shipments. Our test will produce our expected results by examining the conditions to be tested in this process, such as, 'Does this shortest distance pass through the space where a building exists?'

1. **Scope**
   1. **What will be tested.**
      1. Whether the calculated route is crossing only the space that the truck can pass through
      2. Whether the route reach a given destination
      3. If the shipment is loaded, whether it does not exceed the maximum loading weight or volume
   2. **What will not be tested.** 
      1. **No test is conducted assuming a destination outside the 25 by 25 square grid.**
2. **Test Strategy**
   1. **This section describes the approach you will take to performing the tests. There are sections below where you can elaborate on different types of tests. Not all these types of tests will be in every project and some projects might have tests which are not listed below. This would be a good section to describe where the test data is being obtained from. You could also describe the different levels of testing which might be used. For example, testing is often broken up into exploratory testing which attempts to make sure that critical defects are removed before next level of testing begins. After exploratory testing catches some of the big critical defects you can go on to functional testing as the next testing cycle to make sure that all the prime functions of the application are being delivered correctly. You can continue to describe all the test deliverables and what roles are responsible for producing and delivering these. You could also include an estimate of how long it is going to take to do the testing.**  
      3.1. System Test: OS - Windows? Mac? Mobile?  
      3.2. Performance Test: how long does it take application to run  
      3.3. Security Test: prob not applicable, is there a login?  
      3.4. Automated Test: testing framework (Test Manager?/Unit Tests?)  
      3.5. Stress and Volume Test: does it handle multiple simultaneous requests? Debatable, might also not really be applicable – it’s just one session per machine for this console application  
      3.6. Recovery Test: also probably not needed – maybe how to restore the app on failure  
      3.7. Documentation Test //NA  
      3.8. Beta Test //NA  
      3.9. User Acceptance Test: does it do the job that we want it to do? Expected output/functionality?
   2. **You could describe the test design process and give an overview of how it will be conducted. You could provide a broad overview of** 
      1. **how to understand requirements,**
      2. **build a traceability matrix, --- can start building this**
         1. **see template: can be mapped to user stories**
      3. **prepare test cases,**
      4. **and have them reviewed by another member of the quality assurance team.**
3. **Environment Requirements**
   1. This section will typically define the hardware and software environment necessary for the tests to be conducted. This could involve specifying that a test computer is necessary to run the tests in a continuous integration process or it might say that all testing is done on the developers workstations. Test harness is might need to be built to conduct the test or you might be using a pre-existing set of testing tools. All of this needs to be laid out with all its requirements so that the testing environment can be set up before the testing begins.
   2. **Minimum requirement:** functions on Windows OS
4. **Execution Strategy**
   1. this is the section where you will describe how tests 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:
      1. **critical** which cause the system to crash or produce anomalous results,
      2. **high** which causes lack of program functionality and might have a work around,
      3. **medium** which is a bug which D crates degrades the quality of a system but often has a work around to give the desired functionality
      4. **Low** which might be an unclear error message or some other minor error that has minimum impact on functionality
      5. **Cosmetic** which is something that makes the user interface less than optimal but still perfectly functional.
   3. **Test Reporting: automated or manual, or combination of both?**
      1. This action will describe what sort of reports should be produced as a result of testing, how often these reports should be produced, and to whom the reports should be sent. It should give some indication of the contents of the reports and under what conditions the reports are generated. You might say that a manager receives a daily report of the number of tests conducted, passed, and failed with a brief description of the areas being tested and the areas which are failing.
      2. This section could also have details of how the testers are going to feed information back to the project managers so that they can assign developers to fix the bugs. This section can detail the communication to occur between management, the development team, and the quality assurance team.
   4. You can also explain how the quality assurance team we'll be able to interact with the developers and how they will be able to work with the developers to resolve the defects found in the software.
5. **Test Schedule: put down a timeline for testing (ex. By which milestone, which tests should be done?)**
   1. **This is the section where you wrote layout a schedule for the testing and be able to give an estimate of how long the testing will take and approximately when it will be complete.**
6. **Control Procedures**
   1. 6.1 Reviews  
      6.2 Bug Review Meetings  
      6.3 Change Request  
      6.4 Defect Reporting
7. **Functions To Be Tested**
8. **Resources and Responsibilities**  
   8.1. Resources  
   8.2. Responsibilities
9. **Deliverables**
10. **Suspension / Exit Criteria**
11. **Resumption Criteria**
12. **Dependencies**  
    12.1 Personnel Dependencies  
    12.2 Software Dependencies  
    12.3 Hardware Dependencies  
    12.3 Test Data & Database
13. **Risks**  
    13.1. Schedule  
    13.2. Technical  
    13.3. Management  
    13.4. Personnel  
    13.5 Requirements
14. **Tools**
15. **Documentation**
16. **Approvals**

\*\*don’t have to fill out every single thing here – just pick a few that make sense and decide on them as a team