Test Plan Group 10

1. **Introduction**
    a. Test Plan Objectives
        i. The objective of the project is to create a program that contains an algorithm to determine the most efficient way to deliver parcels from our offices, based on predetermined routes. We will be testing through scenarios to ensure that each possible delivery location can be accessed by our trucks and that the trucks are able to take the shortest route to save on exorbitant gas charges.
2. **Scope**
    a. Our focus is to test, within one day, which trucks will hold each package, whether the trucks are able to reach any deliverable location using the shortest paths. If trucks are unable to take the package, we need to ensure the package is set to be shipped out the next day.
    b. The program will not be testing for logistics such as time of departure (aside from the day of shipping).
3. **Test Strategy**
    a. We have test data that will cover the average delivery locations, data that tests the edge cases of delivery locations, and data of varied sized packaging. We will conduct tests as stated below to ensure all aspects of the program function as required.
    b. 3.1 User Acceptance Test
        - The user should be able to easily run the program and be guided on what data to enter
    3.2 Unit Test
        - Program to be broken into units to test functions separately, paying attention to the small details.
    3.3 Integration and End-to-End Test
        - All functions within the program run well with each other and are used efficiently
    3.4 Performance and Stress Test
        - Program should be able to scale and work appropriately with different workloads while retaining stability.
    3.5 Regression and Sanity Test
        - Any new code must pass regression testing to ensure functionality with existing code.
4. **Environment Requirements**
    a. Testing will be done via integrated terminals on Windows 11 as well as on Visual Studio.
5. **Execution Strategy**
    a. Test Criteria
    - Entry criteria:
        + The test will take place after all the required functions are built and the main features like addPackage, calculateCapacity, and calculatePath.
    - Exit criteria:
        + All the tests are considered when:

100% the important tests and seriously passed.

95% all the tests passed that without the serious issues affected to the functions.

b. You can describe the severity of defects in this section and break them down into severity levels of:
   i. **critical**
      - Causing the system issue or creating the incorrect result.
   ii. **high**
      - Causing to loss of the main function but there is still a solution.
   iii. **medium**
      - Causing to decrease in the quality of the system but there is still a solution
   iv. **Low**
      - The small issues that do not affect the main function.
   v. **Cosmetic**
      - The issues do not affect the interface or the function.

c. Test Reporting

- Daily summary reports:
  + Recipient: project manager, developer team, QA team
  + Content: Summarize the tests that were conducted, passed, and failed.
  + Frequency: Everyday
  + Purpose: Provide general about the status of the project, and allow managers to observe and solve all problems
- Defect report:
  + Recipient: Development team and QA team
  + Content: The detail of the serious bug, high or medium that was not solved
  + Frequency: Created when discovering new defects
  + Purpose: Provide the details that can apply, prioritize, and solve the problem effectively for development teams.

i. Feedback and communication:

- Testers will report the bugs by an issue-tracking tool, flagging critical and high-severity issues for immediate attention.
- The project manager will assign defects to developers based on the severity and urgency.
- Testers will retest resolved defects to confirm fixed and update defect status.

6. Test Schedule

a. Testing will take an approximation of 5 weeks, outlined as below:

i. WEEK 1 (Oct. 28 - Nov. 1)
   - Analyze objective
   - Develop prototypes and data structures
   - Create test plan

ii. WEEK 2 (Nov. 4 - Nov 8)
   - Blackbox testing

- Develop C++ testing environment
- Begin implementing functions
- Unit testing
- Minimal integration testing

iii. WEEK 3 (Nov. 11 - Nov. 15)
- Whitebox testing
- Store documentation of passing and failing tests

iv. WEEK 4 (Nov. 18 - Nov. 22)
- Integration testing
- User Acceptance testing
- Regression and Sanity testing

v. WEEK 5 (Nov. 25 - Nov. 29)
- Debugging
- Final test passing
- Complete final report

7. Control Procedures

a. 6.1 Reviews

- This section record the reviews of the test plan, strategy, and cases to make sure meets the requirement of project.

6.2 Bug Review Meetings

- Create the schedule to review the defects and discuss the severity, priority, and assignment for solution.

6.3 Change Request

- Process to request and approve changes to the test plan or project scope

6.4 Defect Reporting

- Details the defect reporting process, including tools, format, and priority levels