Test Plan - Team4

1. **Introduction**
   1. **We will be testing a truck delivery program that uses distance measurement and a delivery algorithm that finds the optimal route to accurately assign packages based on each truck's weight and size capacity, proximity to the destination, and available routes.**
2. **Scope**
   1. **All functions included in the program will be tested.** 
      1. Assigning packages to trucks based on weight and volume capacity
      2. Determine which truck is closest to the destination
      3. Calculate the shortest route from the truck's route to the destination
      4. Handling when a truck can't reach a destination
3. **Test Strategy – included in Github** 
   1. **The detailed test strategy for this project has been documented in a separate file.**
      1. [Summer24-SFT211-NGG-4/Documents/Testing/Tests Documents/ms2-test-strategy-team4.docx](https://github.senecapolytechnic.ca/slee588/Summer24-SFT211-NGG-4/blob/main/Documents/Testing/Tests%20Documents/ms2-test-strategy-team4.docx)
4. **Environment Requirements**
   1. **Hardware**
      1. **Mac OS: Requires a Mac computer with sufficient hardware specifications to run the software.**
      2. **Windows: Requires a Windows computer with sufficient hardware specifications to run the software.**
   2. **Software**
      1. **Mac OS: Perform the test on the latest macOS (e.g., macOS Big Sur), with the required dependencies and software tools installed.**
      2. **Windows: Performs tests on the latest Windows (e.g., Windows 10) and requires that you install the necessary dependencies and software tools.**
   3. **Visual Studio 2022**
      1. **Install on a Windows or Mac OS computer and use for testing and development.**
   4. **Test harness**
      1. **Performs tests using an existing set of test tools. These tools must be installed and properly configured on the test machine.**
5. **Execution Strategy**
   1. Entry Criteria for Testing
      1. The software build to be tested must be ready and deployed on the test machines.
   2. Exit Criteria for Testing
      1. Testing is considered complete when 100% of the test scripts pass and there are no critical or high severity defects remaining.
   3. Severity Levels of Defects
      1. Critical: Defects that cause the system to crash or produce anomalous results.
      2. High: Defects that result in a lack of program functionality and may have a workaround.
      3. Medium: Defects that degrade the quality of the system but usually have a workaround to provide the desired functionality.
      4. Low: Minor errors such as unclear error messages that have minimal impact on functionality.
      5. Cosmetic: Issues that make the user interface less than optimal but are still fully functional.
   4. Test Reporting
      1. Report Creation: Test reports are created to track the progress and results of testing activities.
      2. Frequency: Reports are generated daily and provide an overview of the number of tests conducted, passed, and failed.
      3. Contents: The reports include a brief description of the areas being tested and the areas that are failing.
      4. Recipients: Reports are sent to the Project Manager (Team Leader) and Quality Assurance Manager.
      5. Feedback and Bug Reporting: Testers provide feedback and bug reports to the Project Manager, who assigns the discovered defects to developers through Jira.
      6. Communication Channels: Regular meetings, emails, and issue tracking systems are used to facilitate communication between the Project Manager, Quality Assurance Manager, and Development Team to ensure collaboration in resolving defects.
6. **Test Schedule**
   1. **Testing should occur every week after the source code is updated and should take 15-30 minutes.**
7. **Control Procedures**
   1. Reviews
      1. Regular reviews are conducted to assess the progress of testing activities and ensure alignment with project requirements and goals. These reviews involve team members, the project manager, and other relevant stakeholders. The purpose of the review is to provide feedback, identify gaps or issues in the testing process, and make necessary adjustments to enhance overall quality.
   2. Bug Review Meetings
      1. **Regular meetings are held to discuss and prioritize identified issues or defects. During these meetings, the testing team and development team review reported bugs, determine their severity and impact, and assign responsibility for resolution. Bug review meetings are crucial for maintaining clear communication, tracking bug fixes, and ensuring timely resolution of identified issues.**
   3. Change Request
      1. During the testing phase, change requests may arise for system modifications or improvements. These requests can be based on stakeholder feedback, identified enhancements, or changes in project requirements. The change request process involves documenting the request, assessing its impact, seeking approval from relevant stakeholders, and implementing the approved changes. This process ensures that necessary modifications are properly documented, reviewed, and implemented with minimal disruption to the project timeline.
   4. Defect Reporting
      1. Defect reporting is essential during the testing process. Testers document identified defects or issues using a standardized defect report format. The report includes details such as defect description, steps to reproduce, expected behavior, actual behavior, and supporting attachments. Defects are categorized by severity (e.g., Critical, Major, Minor) and prioritized for resolution. The defect reporting process facilitates effective tracking, communication, and resolution of identified issues.
8. **Functions To Be Tested**
   1. Delivery Assignment Function:
      1. Verify that deliveries are correctly assigned to trucks based on weight, box size, and destination. Test cases cover a variety of scenarios, including valid and invalid inputs, multiple truck availability, and capacity considerations.
   2. Shortest Path Calculation Function:
      1. Validate that the shortest path between two points is accurately determined. Test cases include scenarios with different start and end points, buildings obstructing the paths, and edge cases.
   3. Capacity Calculation Function:
      1. Ensure that the available capacity of each truck is accurately determined. Test cases cover various combinations of weights and box sizes, scenarios where maximum weight or volume is reached, and handling trucks with different constraints.
   4. Output Message Generation Function:
      1. Confirm that the system generates accurate and informative messages regarding truck selection, detour routes, and other relevant information. Test cases address a range of assignment scenarios and edge cases to verify the correctness and clarity of output messages.
9. **Resources and Responsibilities**  
   8.1. Resources
   1. test environment (An appropriate testing environment including necessary hardware, software, and simulation data such as Web manuals and guides, Visual Studio debugging tool)
   2. tester (a dedicated team of testers responsible for executing test cases, documenting results, and reporting any issues or defects),

8.2. Responsibilities

a. Ensure all functions are in working order.

b. testers are responsible for executing test cases, documenting test results, and reporting issues or defects found during testing

c. developers are responsible for resolving reported issues and defects, making necessary code changes, and retesting fixes

d. the project manager is responsible for overseeing testing activities, ensuring proper coordination between the testing and development teams, and providing necessary support and resources to facilitate effective testing.

1. **Deliverables**
   1. Test Plan Document
   2. Test Cases
   3. Test Reports
   4. Defect Logs: Logs detailing identified defects, including severity, steps to reproduce, and status.
   5. Traceability Matrix
   6. Test Data
   7. Quality Assessment Reports
   8. Documentation Updates
2. **Suspension / Exit Criteria** 
   1. **Suspension Criteria:**
      1. **High Percentage of Critical Defects Found:**
         1. **Testing will be suspended if more than 30% of the test cases uncover critical defects. Critical defects are issues that cause the system to crash or produce abnormal results.**
      2. **Lack of Critical Resources:**
         1. **Testing will be suspended if any of the following critical resources are insufficient:**
            1. **Test Environment: Required hardware or software is not available or not functioning properly.**
            2. **Test Data: Insufficient test data to cover the necessary scenarios.**
            3. **Human Resources: Insufficient testers to execute the test cases.**
      3. **Project Scope Changes:**
         1. **Testing will be suspended if changes in project scope affect testing priorities. This includes major additions, deletions, or modifications to key features, requiring adjustments to the test plan.**
   2. **Exit Criteria:**
      1. **High Percentage of Planned Tests Executed Successfully:**
         1. **Testing will be considered complete when 100% of the planned test cases are executed successfully, meaning all test cases produce the expected results.**
      2. **Specific Defect Resolution Targets Met:**
         1. **Testing will be considered complete when all critical and high-severity defects are resolved, ensuring the system is stable and key functionalities are working correctly.**
      3. **Approval from Professor or Project leader:**
         1. **Testing will be considered complete upon the review and approval of test results by stakeholders or project sponsors. This approval confirms that the test results meet project requirements and the product is ready for release.**
3. **Resumption Criteria**
   1. **Availability of Resources:**
      1. **The test environment is fully restored.**
      2. **Required test data is available.**
      3. **Team members are available and prepared to resume testing.**
   2. **Approval Requirements:**
      1. **Approval from project management or stakeholders to resume testing.**
      2. **Verification of changes or fixes implemented during the suspension.**
4. **Dependencies**
   1. Personnel Dependencies
      1. Team Member Availability: Ensuring all team members are available and have sufficient time to participate in testing and development activities.
   2. Software Dependencies
      1. Visual Studio IDE: Required for development, testing, and debugging.
      2. Seneca GitHub: Access to the repository for version control and collaboration.
   3. Hardware Dependencies
      1. Computers: Access to computers that meet the hardware specifications required to run Visual Studio IDE and other necessary development and testing tools.
   4. Test Data & Database
      1. Test Data: Availability of comprehensive and accurate test data to cover all necessary scenarios.
      2. Database Access: Ensuring the database is properly configured and accessible for testing purposes.
5. **Risks**
   1. Schedule
      1. Delays: Potential delays in testing or resolving defects could impact the overall project timeline.
   2. Technical
      1. Algorithm Bugs: Bugs in the shortest path algorithm could lead to incorrect diversions, affecting delivery efficiency and accuracy.
   3. Management
      1. Resource Allocation: Ineffective allocation of resources may lead to bottlenecks and inefficiencies in the testing process.
   4. Personnel
      1. Team Availability: Insufficient availability of key team members could delay testing and development activities.
   5. Requirements
      1. Scope Changes: Changes in project scope or requirements can lead to adjustments in the test plan and potentially delay the project.
6. **Tools**
   1. **Visual Studio IDE: Utilized for development, testing, and debugging of the truck delivery program.**
   2. **Seneca GitHub: For version control and collaboration.**
   3. **Jira: For issue tracking and project management.**
   4. **Microsoft Teams: For team communication and collaboration.**
   5. **Email: For formal communication and report dissemination.**
7. **Documentation**
   1. Test Plan
   2. Test Cases
   3. Test Reports
   4. Defect Logs
   5. Quality Assessment Reports
8. **Approvals**
   1. **Project Manager: Responsible for approving the test plan, test cases, test reports, and overall testing strategy.**
   2. **Team Leader : Final approval of the test results and overall project to confirm readiness for release.**