Test Plan – Group 3

1. **Introduction**
   1. Test Plan Objectives
      1. The objective of this test plan is to ensure that the project’s delivery algorithm accurately assigns packages to trucks based on their capacity, proximity to the destination and available route.
         1. The test plan ensures that all possible testing procedures are completed and all possible test cases are covered for the entire project.
2. **Scope**
   1. Package assignment to trucks based on weight and volume capacity.
   2. Determining the closest truck to the destination.
   3. Calculating the shortest path from the nearest point on the truck’s route to the destination.
   4. Handling cases when trucks cannot reach the destination.
3. **Test Strategy**

|  |  |
| --- | --- |
| **Project Name**  winter25-sft221-nee-group3 (Efficient Shipment Delivery) | **Author**  QA LEADS: Kaitlyn Cassiela Marino, Grace Gabrielle |
| **Computing Environment**  Windows 11, Visual Studio 2022 | **Software Type**  Visual Studio 2022 with GCC compile |
| **User Demographics**  SFT221 class, end users | **Assumptions**  User does not know the functionality except for the specifications described in the Project Overview and milestones. |
| **Purpose of Test**  Fix and improve the given source code and ensure that the program runs and works correctly. | **Phases of Testing**  Unit Testing, Functional Testing, Integration Testing, User Acceptance Testing |
| **Scope of Testing**  **What will be tested or not tested:**  Correct Shipment Distribution, Truck’s Path and Capacity Calculation, and Output Message Functions  The integration of all functions and the functionality of the overall program. | **Critical Success Factors**  **Conditions which indicate success:**  When 95% of tests pass with no high or critical errors that could impact the overall program, when the map/grid is navigated correctly, and when the actual outcome matches the expected outcome in the Project Overview. It should also pass the if-statements according to the milestone assumptions. |
| **Testing Types**  **Types of testing :**  Black Box Testing, Unit Testing, White Box Testing, Functional Testing, Integration Testing, User Acceptance Testing, and Regression Testing | **Tester Profiles**  **Roles to conduct test :**  Varies depending on the capabilities and availability of the team members for each milestone. |
| **Development/Test Tools**  **Tools and environment needed :**  Visual Studio Code 2022 for analyzing, compiling and running the code, creating test cases, testing, and debugging. GCC compiler or CMD interface as needed. | |
| **Business / Operational Concerns**  **Business reasons for the test:**  Member availability and programming experience. Incorrect navigation or difficult bugs can also cause delays in the progress of the overall project. | |
| **Risks**  **Business**  Member availability will vary each week, which may cause delays in the completion of each milestone. Experience with programming, testing, and debugging may also cause delays depending on the severity of the problem.  **Technical**  The GitHub repository or the Jira Project might accidentally be tampered with. Additionally, Visual Studio 2022 might crash during the testing and debugging of the code. The hardware or softwares used by the members might be faulty or insufficient for the project requirements.  **Project**  **Other risks to the project:**  N/A | |
| **Other**  **Any other notes:**  N/A | |

* 1. **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. Understanding Requirements - Testers will thoroughly analyze the project requirements, functional specifications, and any relevant files and documents to gain a clear understanding of what needs to be tested. They will identify the main features, functionalities, and the expected outcome of the program.
     2. **Traceability Matrix – The traceability matrix will act as a reference to track the coverage of all test cases to ensure that all possible features of the program are tested.**
     3. **Preparing Test Cases – The test cases will be made based on the identified requirements of the user/project, as well as the expected output as shown in the project document.**
     4. **Reviewing Test Cases – The test cases will be reviewed by another member of the quality assurance team to ensure that all possible scenarios are covered by the test cases and to provide additional feedback on how to enhance the implementation of the test cases.**

1. **Environment Requirements**
   1. Hardware
      1. Processors like Intel i5 or equivalent are needed to run the program smoothly and without any issues. Additionally, there should be a minimum of 8GB RAM and sufficient amount of storage that can store the source code and the various documents of the project.
   2. Software
      1. Windows OS – The project will require devices that support and use Windows OS to create and update the necessary documentation. It will also serve as the main medium of communication for all group members.
      2. Visual Studio 2022 – The project will require the Visual Studio 2022 software to develop, fix, test, and complete the whole program. This is the main software that will be used by all group members.
2. **Execution Strategy**
   1. Entry Criteria – The program to be tested should be available and can be deployed on demand (with the use of Visual Studio 2022).

Exit Criteria – The tests can be considered complete when 95% of the test cases pass, with no high or critical errors remaining.

* 1. Severity Levels:
     1. **critical** – Errors that cause the system to crash or produce questionable our anomalous results
     2. **high** – Errors that cause insufficient program functionality, but might have a work-around
     3. **medium**– Errors that degrade the quality of the system but often has a work-around
     4. **Low** – Errors that might include unclear error messages or other minor errors that has minimum impact on the overall functionality
     5. **Cosmetic** – Issues that make the use interface less optimal but do not affect functionality
  2. **Test Reporting**
     1. The test reports will be produced to track the progress and results of the testing activities, as well as to keep all team members up to date on what should be done next. The reports will be generated at least three times a week, or when there are errors that still need to be fixed for that week. The reports will consist of the number of tests conducted, passed, and failed. They will also include a brief description of the areas being tested and the areas that are failing.
     2. The testers will send the concise reports to the project manager, who will then assign developers to fix the errors and bugs found in the program according to the failed test cases. All team members involved – such as the developers and quality assurance team – will collaborate and communicate with each other through various online methods and in-person meetings.

1. **Test Schedule**
   1. Testing Estimate: The testing is estimated to take approximately five weeks to complete.
   2. Completion: The testing is expected to be finished by the end of the sixth week from the start of the testing process.
2. **Control Procedures**
   1. **7.1** Reviews – Regular reviews will be conducted to assess the testing process and ensure that the testing procedures align with the project requirements. This involves all team members who will either provide feedback or identify issues in the testing process, and then make the necessary adjustments to improve the overall quality of the system.  
      **7.2** Bug Review Meetings – Regular meetings will be scheduled to discuss and prioritize any identified issues or errors in the program. These meetings can either be held online or in-person, in which all members involved will collaborate to analyze the reported errors, determine their severity, and assign responsibilities to efficiently resolve the error or bug.  
      **7.3** Change Request – Change requests are possible, provided that the team member who proposes these changes have a logical reason to change the current code which could improve the overall program. The change requests will go through the necessary process to document the requested changes, evaluate their impact, seek approval from the project manager or the majority of the team, and then implement the changes to the code.  
      **7.4** Defect Reporting – As mentioned in the Execution Strategy, the testers will document any identified defects or errors through properly-formatted reports. The reports will include brief descriptions of the failed test, the location of the error or bugs, the expected and actual outcomes, and any supporting documents.
3. **Functions To Be Tested**
4. Shipment Distribution Function – This function will be tested to ensure that all shipments are distributed correctly to trucks based on their weight, box size, and destination. The test cases will cover valid and invalid inputs, multiple truck availability, and cases for the truck capacity.
5. Shortest Path Calculation Function – This function will be tested to ensure that it correctly calculates the shortest path to the given destination while simultaneously avoiding walking through buildings. The test cases will cover cases with valid and invalid starting and destination points, buildings blocking the way to the destination, and corner cases.
6. Capacity Calculation Function – This function will be tested to ensure that it accurately determines the available capacity of each truck. The test cases will cover various cases with different weight and box sizes, edge cases for the weight or volume, trucks with specific limitations, and invalid input.
7. Output Message Function – This function will be tested to ensure that the program generates the correct output messages depending on the delivery distribution, shipment weight and size, truck selection, diversion paths, and other relevant information. The test cases will cover different distribution scenarios and combinations, as well as edge cases to validate the accuracy and clarity of the output messages.
8. **Resources and Responsibilities**  
   **9.1.** Resources
   1. **Testers – The testers will create and execute the test cases, document the results, and regularly report any errors or defects.**
   2. **Test Environment – The test environment includes the appropriate hardware, software, and necessary data to be able to support and initiate the testing activities.**

**9.2.** Responsibilities

1. Testers – The testers are responsible for creating and executing the test cases, documenting the results of each test case, and the production of regular reports to summarize the status of the test cases.
2. Developers – The developers will analyze the reported errors and defects, make the necessary code fixes, and notify the testers of the updates.
3. Project Manager/s – The project manager/s will oversee the testing and development of the overall program, ensuring efficient coordination within the team and assigning roles to the respectful team member who are capable of fixing or testing. They will also provide the necessary support and resources to stabilize the flow of the project development.
4. **Deliverables**

The algorithm must accurately allocate packages to trucks, considering available space, distance to the destination, and any necessary detours. It should determine the shortest route from the closest point on the truck's path to the destination. The algorithm should address situations where a truck is unable to reach its destination due to obstacles. It must display the assigned truck, the delivery location, and any detour routes if needed.

1. **Suspension / Exit Criteria**

* If the algorithm is unable to assign packages to trucks considering space, distance, and diversion requirements.
* If the algorithm incorrectly calculates the shortest path or encounters issues in pathfinding.
* If the algorithm fails to handle scenarios where a truck cannot reach the destination due to obstacles.
* If the algorithm fails to display the required information accurately.
* Terminate the algorithm if the baggage weight exceeds 1000 kg.

1. **Resumption Criteria**

After fixing any issues or bugs, the algorithm should be retested to ensure it meets the deliverables. It should be tested with different scenarios and test cases to validate its functionality and accuracy and retested after modifications to confirm that no new issues are introduced.

1. **Dependencies**  
   **13.1 Personnel Dependencies:**  
   We should list the people involved in the project, including their roles and what they’re responsible for. It’s also important to mention if there are any dependencies on their availability or specific skills.

**13.2 Software Dependencies:**  
For testing to work well, we’ll need certain software, like the app being tested, test management tools, bug-tracking systems, and automation tools. The app should be stable, functional, and have all the features it needs. Also, the testing tools must work well with the software to get accurate results.

**13.3 Hardware Dependencies:**  
We should list any hardware needed for the project, like specific components or devices that must be available or connected.

**13.4 Test Data & Database:**  
Having the right test database or environment is key to testing things like data handling, storage, and retrieval. The test data should be realistic, valid, and cover different scenarios, including edge cases like varying package weights, sizes, destinations, etc.

1. **Risks**

**14.1 Schedule:**  
Schedule risks involve issues with meeting testing deadlines, like development delays, scope changes, limited resources, test data availability, and external dependencies. To handle these risks, we need realistic project planning, clear communication with stakeholders, effective prioritization, and proactive risk management.

**14.2 Technical:**  
Technical risks can affect the quality and efficiency of testing, like infrastructure issues, network problems, hardware limitations, test environment setup challenges, tool constraints, and data management issues. To reduce these risks, we should ensure compatibility, set up the environment properly, and manage data effectively. Working with experts and staying up to date on tools and security can also help minimize these risks.

**14.3 Management:**  
Management risks are related to how the project is coordinated and how it progresses. Poor project management can delay timelines and affect deliverables. Identifying these risks early and having strong project leadership can keep things on track.

**14.4 Personnel:**  
Personnel risks within the team include turnover, low motivation, communication problems, skill gaps, and conflicts. We can reduce these risks by building a positive team culture, encouraging open communication, offering support, and handling conflicts quickly. Regular meetings and offering support can help build team cohesion and individual satisfaction.

**14.5 Requirements:**  
Risks related to unclear, incomplete, or changing requirements can affect the project’s scope and deliverables. We should regularly review and clarify the requirements to make sure they align with the project goals and minimize disruptions caused by scope changes.

1. **Tools**

Testing tools are key to making sure our testing is efficient and accurate. These tools include test management tools to help us plan and track tests, defect tracking tools to handle software issues, performance testing tools to check how well the system performs, and test data management tools to generate the test data we need. Choosing the right tools can really boost our testing efficiency and productivity.

1. **Documentation**

During testing, we generate different documents like test cases, test scripts, test reports, defect logs, and user guides. Each one has its own purpose and gives us useful info. The format and structure can vary, and there might be specific templates or guidelines we need to follow. It’s important for us to understand these templates and stick to them so we can document the testing process properly and communicate the results clearly.

1. **Approvals**

The individuals responsible for approving the test plan and associated documents typically include project managers, QA leads, technical leads, and key team members like product owners. The approval process involves preparing the test documents, conducting internal reviews, gathering feedback from team members, and making necessary revisions. Once all feedback is addressed, the final documents are submitted for approval. The approval process should be completed within 2-4 business days per review stage, ensuring all parties have reviewed and consented before testing activities begin.