Test Plan Template

1. **Introduction**

The Program:

**The project involves creating a program that determines the routes of three trucks that deliver packages to given addresses. The packages that the trucks can receive have specific requirements, as well as the volume and weight that the trucks can carry at once. The program’s objective is to find the optimal route for all three trucks so that as many packages can be delivered in one day with the trucks going in the most efficient path.**

Test Plan Objectives:

**The testing will be carried out in different stages as it will be carried out in different stages of the program’s progress. The tests involve whether the program will run on different operating systems. Then the different functions within the program can be tested for their performance through modular/unit testing. Automated tests will also be used to check numerous conditions to see if diverse requirements are met. The tests will ensure that the program properly assigns packages to the trucks and that the most efficient paths are chosen for each truck.**

1. **Scope**

**Testing may include checking whether the program is compatible on both windows and mac**

**operating systems. The main business requirement to test will be to see if the routes**

**assigned did go on the most efficient paths without going through Buildings.**

**Testing will not include how long the routes may take time wise or confirm whether the**

**packages were successfully delivered or not.**

**Load and stress testing may take place to check the amount of data the program may**

**handle, but the program is not a web application, meaning the number of users who can**

**access the program will not be tested.**

1. **Test Strategy**
   1. **As different parts of the program will be implemented as the program is being developed, different types of testing will be integrated throughout the progression of the project. Each function will be tested in the program to ensure that the expected behavior is being executed properly through unit tests. Then, integration tests can be carried out to check that the functions are compatible with each other and still give the wanted outcome.**   
        
      **The test strategy will take place in the following sequence:**  
       **3.1. Unit Testing**  
       **- black box test data (MS3)**  
       **- white box test data**  
       **3.2 Integration Testing**  
       **- ensure that different units are compatible with one another**  
       **- Functional Testing: ensure that the output meets the relevant business**   
       **requirements**  
       **3.3 End-to-End Testing**  
       **- ensure that the entire program can run smoothly from start to finish**  
       **3.4 Load & Stress Testing**  
       **- testing to ensure the program may handle large order amounts**  
       **3.5 Acceptance Testing**  
       **- ensuring all business requirements are met**
   2. **Each test design process will take place in the following order**
      1. **The function specifications will be declared,** 
         * **This will include what the function does, unusual conditions, and parameters**
      2. **Build a traceability matrix (excel sheet),**
      3. **Prepare test cases,** 
         * **Test cases will involve test data that is based off both black box and white box testing.**
      4. **The tests are reviewed by another member of the quality assurance team.**
2. **Environment Requirements**

The testing for this program requires a windows operating system capable of running Visual Studio debugger. Testing will be carried out continually throughout the integration process, and the Visual Studio debugging system will be applied to run the tests.

1. **Execution Strategy**
   1. In the case of unit testing, integration testing, end-to-end testing, and load & stress testing the tests can be passed and the next step of testing can be entered only if there are defects no higher than ‘Low’. In certain cases, having one or two ‘Medium’ severity levels may be permitted to exist depending on the priority of the bug.  
      Any testing that checks the business requirement will only be passed if all user requirements are met.
   2. The severity levels break down as:
      1. **Critical** - causes the system to crash or produce anomalous results.
      2. **High** - causes lack of program functionality and might have a work around.
      3. **Medium** - a bug which degrades the quality of a system exists but often has a work around to give the desired functionality.
      4. **Low** - may have an unclear error message or some other minor error that has minimum impact on functionality.
      5. **Feature Request** - is something that makes the user interface less than optimal but still perfectly functional.
2. **Test Reporting**
   1. A function report shall be created for each step of testing, along with a traceability matrix. These reports should be uploaded to GitHub and communicated to the rest of the team so that the appropriate team members may access and react accordingly.
      1. The function report will pertain test cases along with any bugs found so that the developers can know where the bug occurred. Tests will be implemented at least once a week. Once initial testing has been completed and then the developer team has fixed the bugs, another set of tests should be executed within 2 days.
   2. All finalized documents shall be shared through GitHub, any documents that need to be worked on together may be shared on Microsoft Teams. Tasks and their status should be updated in a timely manner on Jira. Any communication can be done through Microsoft Teams.
3. **Test Schedule**
   1. **Testing times may vary depending on the milestone of the project; however, each set of tests will be aimed to finish within 2 days of beginning the tests. As the tests depend more on developing test cases to run on Visual Studio, they should ideally be completed within one day. This should give the developers more time to fix bugs when they occur. Once bugs are believed to be fixed, tests will resume with updated test cases if necessary, and once again aim to finish within a day.**
4. **Control Procedures**

* Review/ Documentation:
* Reviews will be routinely conducted once a week and will include looking over documentation such as the test plan, test cases, test code, as well as the traceability matrix to ensure the accuracy and adherence to the requirements.
* Test Environment:
* The test environment will closely resemble the production environment; both test and production will be conducted using Visual Studio.
* If the production environment changes for any reason throughout the testing process, the test environment will also reflect those changes to maintain similarity between the two environments.
* Bug Review Meetings and Progress Monitoring
* Bug review meetings will be conducted to discuss and prioritize reported defects. The meetings should be conducted soon after the defects are reported so that there is ample time to fix the bugs.
* Bug fixes and resolutions will be tracked and verified during subsequent testing cycles.
* Defect Reporting
* Defects will be reported by indicating which function and which test case did not properly meet the requirements. Therefore, the test description reports will be used to document defects and should be communicated to the developer team through MS Teams as well as creating a task on Jira.
* Risk Management
* The workload of each work cycle should consider the amount of time each task may take and be distributed to each member to ensure the tasks can be completed in a timely manner.
* The status of each task, what has been completed and what tasks remain to be completed should be regularly completed.
* If a member assesses that they are unable to finish a task within the designated deadline, then this should be communicated right away to group members to mitigate the task and reduce any delays.
* Issues that arise within the group, whether it is with the code or member disputes, should be discussed in meetings so that the best order of action can be agreed upon before implementation.

1. **Functions To Be Tested**

The following functions will be tested from the utils module from the source code:

* int validate(int weight, double volume, struct Point valid);
* double getSpaceRemaining(const struct Truck\* truck);
* struct Truck\* getTruckByReference(struct Fleet\* trucks, int routeSymbol);
* void getTruckDistances2(double arr[][2], struct Fleet\* current, struct Point destination);
* void sortByLimitingFactor(double dists[][2], struct Dispatch\* org);
* int findTruckAndDiversion(struct Dispatch\* org, double dists[][2], struct OrderInfo\* order);

\*If changes are made to these functions prototypes or functions are added or removed,  
 then this list will be updated appropriately.

1. **Resources and Responsibilities**

|  |  |
| --- | --- |
| **Member** | **Roles and Responsibilites** |
| **Cesca Dela Cruz** | **Tester** |
| **Gulpreet Kaur** | **Developer** |
| **In Tae Chung** | **Tester** |
| **Irish Banga** | **Developer/Tester** |

1. **Deliverables**  
   **In this project, the deliverables will be the Test Plan, the Traceability Matrix, Test Cases (and test data), the Test Report, and the actual source code for the final project.**  
   **These deliverables will be considered successfully delivered when uploaded to GitHub.**  
   **There should also be clear indication of which tasks are yet to be started, which tasks are in progress, and which tasks are completed through Jira.**
2. **Suspension / Exit Criteria**

**A temporary suspension may occur if any unforeseen circumstances occur that effects the group members’ roles and responsibilities (i.e., A group member is unable to contribute to their given roles). If the suspension is to occur, the professor will be notified immediately to ensure that the project may resume in an orderly fashion.**  
**The project will be permanently concluded only after the day of final submission of the project, if any part marks may be gathered through submission, the project will resume.**

1. **Resumption Criteria**

**Only the professor may decide to suspend or exit the project, but should any unforeseen circumstances occur affecting the roles and responsibilities of group members, the professor should be notified right away.**

**The testing shall resume once all the testing roles can be completed by a group member.**

**Any communication related to suspension and resumption of this project will be done along with the professor.**

1. **Dependencies**

14.1 Personnel Dependencies

* Developers will be required to develop source code as well as fix any bugs
* Testers will be required to develop test cases and implement tests to ensure requirements are being met.
* All members shall bear the responsibility of attending weekly meetings and ensuring their individual tasks are being completed by their set deadlines.

14.2 Software Dependencies

* All members are required to use GitHub and Jira to ensure accuracy of completed tasks and documentation
* Visual Studio will be used to develop both source code and test cases
* Microsoft Teams will be used to maintain a convenient method of communication.

14.3 Hardware Dependencies

* Member’s computers should ensure that they are able to run the above softwares.

14.4 Test Data & Database

* Test data should be accurately documented on each functions’ test definition document.
* Accurate descriptions of developed functions will also ensure the accuracy of test data.
* Recorded test cases should also be reflected on the traceability matrix to ensure an overall view of how business requirements are being met.

1. **Risks**

15.1. Schedule

* After reviewing what each milestone entails, group members will engage in weekly meetings to discuss the distribution of tasks.
* As tasks become increasingly time-consuming, additional meetings may be required to ensure all members are aware of which tasks need to be prioritized.
* If tasks are only dependent on select members, a meeting may not require all members
* Any tasks that may not be completed by a deadline should be indicated right away so that the best course of action can be agreed upon by other group members.

15.2. Technical

* As program development as well as testing are time-consuming tasks, group members should be prepared to change roles between developers and testers to ensure that bottlenecks are reduced as much as possible

15.3. Management

* If communication is ineffective, roles are inadequately completed, or there is a lack of engagement from a group member, the issue should be clearly communicated to the individual to ensure the problem can be avoided in the future. If the issue persists, the professor should be notified to ensure individual marks are not jeopardized.

15.5 Requirements

* Each member should have a clear idea of what deliverables are required for each milestone, and work to best accomplish deliverables assigned to them. If possible, group members should also be proactive in helping complete tasks that are yet to be finished.
* As the project develops, different requirements may need to be prioritized. The severity and priority of each requirement should be discussed within the group to ensure there is a consensus on what should be prioritized. This will work to ensure there is clear communication between members and help lessen the probability of conflicts occurring.

1. **Tools**

**In this project we will use Jira and GitHub for tracking and uploading project progress.**

**Microsoft Teams will be used to ensure proper and timely communication between group members.**

**Visual Studio will be used to write and debug the source code of the program.**

**Microsoft Word and Excel spreadsheets will be used to create documentation and reports.**

1. **Documentation**

**In this project the testing process requires the test plan, the function specification reports, and the traceability matrix.**   
**All documents shall be uploaded to GitHub so that all members may see the progress and execute their roles efficiently. The tasks will also be recorded through Jira so that each member can keep track of what needs to be completed.**

1. **Approvals**
   1. **All tests are required to have the definition and test case reports reviewed by a member of the group. It is every group member’s responsibility to communicate to the rest of the members if any of the reports seem to be incomplete or contain errors.**   
      **Other documentation that needs to be constantly updated and reviewed include the traceability matrix.**
   2. **Any reports created or updated during a week will be reviewed by group members prior to the milestone submission to ensure that everything is in proper working order and maintain quality assurance. Another way to ensure that things are updated properly is by checking the Jira Kanban board to see that all the tasks were completed properly as well.**