Test Plan Template

* **The test plan may be updated to better encompass how the tests are built**
* **The highlighted sections are identified for easier removal later on**

1. **Introduction**

**This section should describe the project that is being tested and what are the objectives for**  **the test. You might give a quick overview of the project to tell people what it's about, and**  **then describe the testing at a very high level and what do you expect to get because of the**  **testing.**

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**

**In this section you can describe what will be tested and what will not be tested.**

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

**operating systems, as well as seeing if the routes assigned did go on the most efficient paths**

**without going through Buildings.**

**Testing will not include whether the trucks overlap on their routes 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. **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 the 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

3.2. Performance Test  
 3.3. Security Test  
 3.4. Automated Test  
 3.5. Stress and Volume Test

3.6. Recovery Test  
3.7. Documentation Test  
3.8. Beta Test  
3.9. User Acceptance Test

* 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 project.**   
     **Each function will be tested in the program to ensure that the expected behavior is being executed properly. 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**  
      **- 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.**

1. **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.

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 developer’s workstations. Test harness might need to be built to conduct the test or you might be using a pre-existing set of testing tools. All this needs to be laid out with all its requirements so that the testing environment can be set up before the testing begins.

1. **Execution Strategy**
   1. This is the section where you will describe how the tests are 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 that a test is completed if there are no severe or critical defects.
   2. 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.
   3. 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.
   4. **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.
         * 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.
      2. 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.
      3. 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.
   5. 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.
2. **Test Schedule**

**This is the section where you write down 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.**

1. **Control Procedures**

This section outlines the processes and practices that will be followed to manage and control the testing activities throughout the software development lifecycle. It also ensures that the testing process is well-organized, standardized, and effectively managed to achieve the desired test objectives within the allocated resources and timeframe. You could include some of these sections:

* Reviews
* Test Environment
* Bug Review Meetings and Progress Monitoring
* Change Request
* Defect Reporting
* Test Execution Schedule
* Documentation
* Risk Management
* Communication

1. **Functions To Be Tested**

This section will be updated as the function specifications are set.

This section lists all functions and features of the software that will be tested.

1. **Resources and Responsibilities**

|  |  |
| --- | --- |
| **Member** | **Roles and Responsibilites** |
| **Cesca Dela Cruz** | **To be deteremined** |
| **Gulpreet Kaur** | **To be determined** |
| **In Tae Chung** | **To be determined** |
| **Irish Banga** | **To be determined** |

**This section will be updated within the next week, by July 13th.**

This section should have all team members listed, their roles and responsibilities associated with the testing activities. It ensures that the necessary resources are allocated appropriately and that team members understand their assigned tasks and responsibilities.

1. **Deliverables**

**This section lists the specific documents, artifacts, or outputs that will be produced as part of the testing process.**   
**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.**

1. **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.**

**This section outlines the conditions or criteria under which testing activities can be temporarily suspended or permanently concluded. These criteria help guide the decision-making process and provide a clear indication of when to pause or complete the testing efforts.**

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.**

**This section outlines the conditions or criteria that must be met for testing activities to resume after a temporary interruption or pause. You could include:**

* **Specific circumstances or events that initiate a change.**
* **Who has the authority to make the decision to suspend testing.**
* **Specific conditions or criteria that must be met for testing activities to resume.**
* **Escalation path and responsibilities in case resumption criteria cannot be met within a specified timeframe some.**

1. **Dependencies**

12.1 Personnel Dependencies

12.2 Software Dependencies

12.3 Hardware Dependencies

12.3 Test Data & Database

1. **Risks**

13.1. Schedule

13.2. Technical

13.3. Management

13.4. Personnel

13.5 Requirements

1. **Tools**

**In this project we will use Jira, GitHub, Excel, and Visual Studio.**

**This section outlines the various software tools and technologies that will be utilized during the testing process.**

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.**

**This section outlines the various documents required throughout the testing process. In this project, the testing process requires the test plan, ...**

1. **Approvals**

**This section outlines the approvals required throughout the testing process. This section ensures that the necessary documentation is prepared, reviewed, and approved to maintain proper communication, traceability, and quality assurance.**