Test Plan Template

1. **Introduction**
   1. Test Plan Objectives
      1. **This project is to make a program with which the user will be prompted for their shipment information and it will decide the truck and shortest route by which the shipment will be delivered, based on the information and current truck availability.**
      2. **Testing is to achieve confidence for the stability and its proper working, which means it will work properly under almost any circumstance.**
2. **Scope**
   * 1. **What will be tested:** 
        + 1. **Functions that changes or creates routes**
          2. **Functions that implements routes the maps**
          3. **Functions that displays the map.**
          4. **Functions that supports other functions(Utility functions).**
     2. **What will not be tested:** 
        + 1. **Functions that just returns a data member.**
     3. **Test the program.**
3. **Test Strategy**
   1. **Description on the approach when performing the tests.**   
      3.1. BlackBox Test  
      3.2. Whitebox Test  
      3.3. Unit Test  
      3.4. Integration Test  
      3.5. User Acceptance Test

3.6. Regression Test

* 1. **Test design process.**
     1. **how to understand requirements : compare each member’s understanding of what the business requires.**
     2. **build a traceability matrix -> list business requirements(each functionality, e.g drawing a route on a map) and design tests to meet the requirements.**
     3. **prepare test cases -> test cases for each testing.**
     4. **and have them reviewed by another member of the quality assurance team.**

1. **Environment Requirements**
   1. Windows OS, Visual Studio 2022 Testing Framework.
2. **Execution Strategy**
   1. Describe the severity of defects in this section and break them down into severity levels of:
      1. **critical** which cause the system to crash or produce anomalous results,
      2. **high** which causes lack of program functionality and might have a work around,
      3. **medium** which is a bug which D crates degrades the quality of a system but often has a work around to give the desired functionality
      4. **Low** which might be an unclear error message or some other minor error that has minimum impact on functionality
      5. **Cosmetic** which is something that makes the user interface less than optimal but still perfectly functional.
   2. **Test Reporting**
      1. The team leader would get reports and will decide whether more testing is required or whether some other codes should be implemented. Reports will be produced and sent before the weekly meeting. And then, the whole team member will discuss the bugs and reports, and then the leader will distribute works to members.
      2. Quality assurance team will post the reports on Jira in a document file and code file, but also send messages on team messenger such as Microsoft Teams just in case the leader and team members miss it.
   3. How the quality assurance team will interact with the developers and how they will be able to work with the developers to resolve the defects found in the software.
      * + 1. Make a group chat on Discord just for developers and quality assurance team, turn notification on, frequently checking the chat. Let each other’s know the progress.
          2. Discussion through weekly meeting, regarding the progress of the team.
3. **Test Schedule**
   1. **The first Unit Test should be done before Meeting day.**
   2. Bug Review Meetings will be done on Meeting (every Monday).
   3. **After reporting bugs and discussing the problems. <= English weird, please fix it later.**
   4. **Then, another test will be done and reported to the leader.**
   5. **After confirmation of the leader, it will be submitted.**
   6. **Integration test completion would be April 5**
   7. **Acceptance test completion would be April 12**
   8. **The final submission would be April 21.**
4. **Control Procedures**
   1. 6.1 Reviews  
      6.2 Bug Review Meetings will be done on Meeting (every Monday)  
      6.3 Change Request  
      6.4 Defect Reporting
5. **Functions To Be Tested**

**i) Functions calculating routes.**

**ii) Functions drawing routes the maps.**

**iii) Functions displaying the map.**

**iv) Functions supporting helper functions.**

1. **Resources and Responsibilities**  
   8.1. Resources : Jira, Github, Discord   
   8.2. Responsibilities

**i) Project Manager : Richard**

ii) Developer : Hyunjin Shin

iii) Quality assurance team : Yelda

iii) Debugger : Jeremy

1. **Deliverables**

**March 8, 2023  
 i) Completed team contract**

**ii) Initialized Git repository**

**iii) Set up Jira project**

**March 15, 2023**

1. **Create a test plan**
2. **Create a series of data structures created as header files and stored in the repository**
3. **Analysis of the problem**

**March 22, 2023**

1. **Set of function specifications stored in the repository**
2. **A set of blackbox tests as test documents with test data for the functions**
3. **Start writing blackbox test code and store in repository**
4. **Start implementing functions and store in repository**
5. **A function-test matrix added to the repository**
6. **Update Jira to show activities and progress**

**March 29, 2023**

1. **Implemented Functions**
2. **Implement blackbox tests and store in repo, execute and add results to Jira, and debugged**
3. **Whitebox tests written and stored in repository.**
4. **Implement whitebox tests and store in repo, execute and add results to Jira, and debugged**
5. **Updated function-test matrix stored to the repository**

**April 5, 2023**

1. **Integration tests written and stored in repository**
2. **Integration tests written and store in repo, execute and add results to Jira, and debugged**
3. **Acceptance tests written and stored in repository**
4. **Updated function integration-requirements-test matrix stored to the repository.**

**April 12, 2023**

1. **Execute acceptance tests and add results to Jira, debug**
2. **Update function-test matrix stored to the repository.**
3. **Final testing report listing tests conducted, bugs fixed and the final test passed**
4. **Suspension / Exit Criteria**

Verify if All tests planned have been run. Verify if the level of requirement coverage has been met. Verify if there are NO Critical or high severity defects that are left outstanding. Verify if all high risk areas are completely tested. Verify if software development activities are completed within the projected cost. Verify if software development activities are completed within the projected timelines.

1. **Resumption Criteria**

When there is an update, or a bug is found.

1. **Dependencies**  
   12.1 Personnel Dependencies : C Language Developers, Debugger, UX Designer, Project Manager.

12.2 Software Dependencies : Windows OS, Visual Studio 2022 Testing Framework, Jira, Github  
12.3 Hardware Dependencies : Intel Computer.   
12.3 Test Data & Database : Given map and truck routes

1. **Risks**  
   13.1. Schedule : When deliverables are not handed on time. Miscalculate the amount of time for work.  
   13.2. Technical : hardware failures, compatibility issues, security vulnerabilities, or inadequate testing.  
   13.3. Management : Several factors, such as poor planning, poor communication, a lack of resources, or insufficient monitoring and control, can lead to management hazards.  
   13.4. Personnel : A member lacks a proper skill required for their job.   
   13.5 Requirements : sudden changes of requirements that will leave dissatisfaction to the client.
2. **Documentation**

**Contracts, Testing plan, test traceability sheet, test strategy.**

1. **Approvals**

Clients: Robert Robson

Project Team: SFT221 - 3

1. Richard

2. Jeremy

3. Hyunjin

4. Yelda