Test Plan – Group 5

**1. Introduction**

a. Test Plan Objectives

Verifing the functionality, performance, and security of the mapping software are the main objective of this plan. The test plan outlines the approach to testing, test deliverables, and responsibilities of each team member involved in testing.

**2. Scope**

1. To identify the shortest route between two points and the best path to go from one point to another.
2. To find the best path with avoiding all obstacles on the way.

**3. Test Strategy**

a. Approach to Testing

The testing approach for the mapping software application includes the following types of testing:

System testing

Performance testing

b. Test Design Process

we'll find what needs to be tested, like the software or app, its different parts, and then, we'll decide which tests to do like checking if it works correctly and making sure if it still works after the changes, and seeing the speed.

Afterwards, we'll choose the tools and things we need for testing, like software, doing it by hand, and managing the tests.

It will also need a plan for the tests and write down what we expect to happen. Then, we'll run the tests and write down actual outputs.

After the results we’ll find any problems that came up during testing and figure out why those problems happened.

Finally, fixing all issues, making sure the data is correct, and improving the testing process for the next time.

**4. Environment Requirements**

Hardware Requirements:

* A hard disk
* At least 4 GB of RAM
* A graphics card
* A computer with a 64bit processor

Software Requirements:

* compatible version of a web browser
* Microsoft Visual Studio
* compatible operating system

**5. Execution Strategy**

1. To perform tests in Visual Studio, you start by opening the Test Explorer and adding your test cases. You also set some rules for each test, like how well it should perform or not have any major problems.
2. You run these tests and see how they do. May use special tags to set the rules for each test, see the results in the Test Explorer.
3. For rating quality use the Test Explorer and the Test Results window. You can check which tests failed and see why they failed, like what caused the issue.
4. You can then sort these issues into different levels based on how serious they are. Critical issues are really bad, making the system crash or act strangely. High issues are serious but may have some workarounds. Medium issues affect the system's quality but can often be worked around. Low issues barely affect the system's performance, and cosmetic issues are just about making things look better but don't affect how the system works

**6. Test Schedule**

This section includes outlining the schedule for testing process and providing an estimate of how long the testing will take to be completed. we need to consider several things such as the complexity of project, the number of tests needed, and the availability of team members who will conduct the tests.

I plan to start by breaking down the testing process into smaller, more manageable tasks. This will help us prioritize the tests based on their level of complexity and importance, and create a timeline for when each task will be performed. It's important to anticipate any potential roadblocks or delays that may arise during the testing process, such as unexpected bugs or issues with resources, so we can plan accordingly.

Once the testing schedule is created, we will communicate it with our team and stakeholders to ensure everyone is aware of the timeline and expectations for the testing process. We will regularly update the schedule as necessary to ensure that we stay on track and meet our deadlines.

By carefully planning and executing our testing schedule, we can ensure that our project is successful and meets all of the necessary requirements."

Top of Form

**7. Control Procedures**

**7.1 Reviews:**

Reviews are an essential part of any testing process, as they help identify potential issues or areas for improvement before they become major problems. In this section, we will outline the different types of reviews we will conduct, such as code reviews, design reviews, and documentation reviews. We will also define the review process, including who will conduct the reviews, how often they will be conducted, and how the results will be documented.

**7.2 Bug Review Meetings:**

Bug review meetings are an important aspect of our testing process, as they help ensure that all bugs and issues are properly identified and addressed. In this section, we will outline the process for bug review meetings, including who will attend the meetings, how often they will be held, and how the results will be documented. We will also define the criteria for prioritizing and resolving bugs, and how the bugs will be tracked and monitored.

**7.3 Change Request:**

Change requests are requests for modifications to the original project scope or requirements. In this section, we will outline the process for submitting and reviewing change requests, including who can submit them, how they will be reviewed, and how they will be approved or rejected. We will also define the criteria for prioritizing and implementing change requests, and how they will be tracked and documented.

**7.4 Defect Reporting:**

Defect reporting is a critical part of our testing process, as it helps identify and document any defects or issues in the software. In this section, we will outline the process for defect reporting, including how defects will be identified, who will report them, and how they will be tracked and documented. We will also define the criteria for prioritizing and resolving defects, and how they will be monitored and reported to stakeholders.

By defining these processes and procedures, we can ensure that our testing process is efficient, effective, and thorough. This will help us identify and address any issues or defects in the software before it is released to our users, ensuring a high-quality product and a positive user experience.

**8. Functions to be tested**

A function specification should contain:

• The name of the function

• A description of what the function does

• A description of any unusual conditions

• The return type and parameters

**9. Resources and Responsibilities**

To make sure we have all the things we need for testing, we'll make a plan that says what resources we need, when we need them, and who's in charge of getting them. This plan will help us make sure we have everything we need when we need it, and that we don't spend too much money or take too long to finish.

**10. Deliverables**

To ensure the quality of our deliverables, we will establish a process for reviewing and approving them, including who will review them, how they will be reviewed, and how feedback will be incorporated. This will help us ensure that our deliverables meet the necessary quality standards and accurately reflect the results of our testing process.

**11. Suspension/Exit Criteria**

This includes factors such as unexpected test results, lack of resources or time, or other factors that could impact the effectiveness of our testing process and changes in project scope or requirements. When suspension criteria are met, we will pause our testing process until the issue is resolved or additional resources are obtained.

The exit criteria include factors such as achieving a minimum level of test coverage, reaching a predetermined number of test cycles, or defect-free runs. When exit criteria are met, we will finalise our testing process and move further.

**12. Resumption Criteria**

The resumption criteria may include factors such as resolving the issues that caused the suspension, acquiring necessary resources or support, re-establishing a stable environment, and re-validating any impacted test cases. When resumption criteria are met, we will restart our testing process with a focus on the areas that were impacted by the suspension.

**13. Dependencies**

Personnel dependencies may include factors such as team member availability, skillset, or workload. It is important to ensure that we have the necessary personnel resources to conduct our testing activities and that everyone involved in the testing process has the appropriate skills and experience.

Software dependencies may include factors such as operating system requirements, compatibility with third-party software, or availability of software licenses. We must ensure that we have the necessary software resources to conduct our testing activities and that they are compatible with the system under test.

Hardware dependencies may include factors such as availability of hardware resources, compatibility with the system under test, or performance of the hardware. We must ensure that we have the necessary hardware resources to conduct our testing activities and that they are compatible with the system under test.

Test data and database dependencies may include factors such as availability of data, compatibility with the system under test, and privacy or security concerns. We must ensure that we have the necessary test data and database resources to conduct our testing activities and that they are compatible with the system under test.

**14. Risks**

13.1 Schedule: Schedule dependencies refer to any factors that may impact the timeline for completing our testing activities. This may include factors such as availability of resources, unexpected delays or issues, or changes to project priorities. It is important to monitor our testing schedule closely to ensure that we are on track to meet our deadlines and make any necessary adjustments if we fall behind.

13.2 Technical: Technical dependencies refer to any factors related to the technology or tools used in the testing process. This may include factors such as compatibility with different operating systems, availability of software or hardware resources, or technical limitations of the system under test. It is important to ensure that we have the necessary technical resources to conduct our testing activities and that they are compatible with the system under test.

13.3 Management: Management dependencies refer to any factors related to the management and oversight of the testing process. This may include factors such as availability of funding or resources, support from management or stakeholders, or changes to project priorities. It is important to ensure that we have the necessary support and resources to conduct our testing activities and that we are aligned with the overall project goals and objectives.

13.4 Personnel: Personnel dependencies refer to any factors related to the availability or skills of the individuals involved in the testing process. This may include factors such as team member availability, workload, or skills and experience. It is important to ensure that we have the necessary personnel resources to conduct our testing activities and that everyone involved in the testing process has the appropriate skills and experience.

13.5 Requirements: Requirements dependencies refer to any factors related to the requirements or specifications for the system under test. This may include factors such as changes to requirements, unclear or incomplete requirements, or conflicts between different requirements. It is important to ensure that we have a clear understanding of the requirements for the system under test and that we are able to effectively test against them.

**15. Tools**

Visual Studio

GitHub

Jira

**16. Documentation**

Test reports

Log File (If required)

Final output of the project

**17. Approvals**

By Team Leader and other team members