**TrailSeek Manual Testing Plan**

Manual testing, refers to a test process in which we manually tests the software application in order to identify bugs. To do so, we will follow a written test plan that describes a set of unique test scenarios. It required to analyze the performance of the web or mobile application from an end user’s perspective.

We will verify the actual behavior of software against expected behavior, and any difference is reported as a bug.

Let’s take a simple example to explain this. A Team member has created login page and wants to test it for functionality. In this case, the expected behavior is that the user must be able to enter the username and password and submit the credentials by clicking the **Login** button.

However, when the test is executed, the Login button doesn’t redirect the user to the home page. In such a case, we will report a bug to the respected member.

Manual testing is an essential part of any test strategy as it helps us gain deeper insight from an end user’s perspective. Since manual testing is carried out by a human without the intervention of test automation frameworks, it judges software from the most important metric:

User Experience.

Manual testing plays a vital role in exploratory testing or in test cases that are executed once or twice. This helps us to discover bugs in the early stages of the development cycle.

Stages of Manual Testing

1. Unit Testing

Unit Testing involves verification of individual components or units of source code. A unit can be referred to as the smallest testable part of any software. It focuses on testing the functionality of individual components within the application. It is often used by developers to discover bugs in the early stages of the development cycle.

A unit test case would be as fundamental as clicking a button on a web page and verifying whether it performs the desired operation. For example, ensuring that a login button on application lets you login the correct page.

2. Integration Testing

Integration testing is the next step after unit testing. Multiple units are integrated to be tested as a whole. For example, testing a series of pages in a particular order to verify interoperability.

This approach helps us to evaluate how several components of the application work together to provide the desired result. Performing integration testing in parallel with development allows developers to detect and locate bugs faster.

3. System Testing

As the name suggests, system testing involves testing all the integrated modules of the software as a whole. It helps us to verify whether the system meets the desired requirements. It includes multiple tests like validating output based on specific input,  
testing user experience and more.

Teams perform several types of system testing like regression testing, stress testing, functional testing and more, depending on their access to time and resources.

4. Acceptance Testing

The main goal of acceptance testing is to verify whether the system as a whole is fit for use in the real world.

Acceptance testing is performed both internally and externally. Internal acceptance testing (also known as alpha testing) is performed by the members within the Team. External testing (also known as the beta testing) is performed by a limited number of actual end-users. This approach helps teams evaluate how well the product satisfies the user’s standards. It also identifies bugs in the last stage before releasing a product.

Types of Manual Testing

1. White Box Testing

White box Testing, also known as glass box or transparent testing, is an approach in which the Team is familiar with the internal code or structure of the application. It is primarily used for unit testing. White box Testing also covers specific techniques like data flow testing, control flow testing, decision coverage, and path testing, and a few others.

2. Black Box Testing

Black-box testing is a test approach in which the QA doesn’t have any knowledge about the underlying code or structure of the application. The QA interacts with the software application just like an end-user to test its functional and non-functional behavior. This helps to discover some bugs typically overlooked in the earlier stages.

3 Grey Box Testing

Grey-Box test approach is the combination of both white box and black box testing techniques. The main aim of this approach is to identify any bugs present either due to inappropriate usage or any structural flaws.

How to do Manual Testing

Here’s how to perform manual testing step by step:

1. Analyze requirements from the software requirement specification document
2. Create a clear test plan
3. Write test cases that cover all the requirements defined in the document
4. Get test cases reviewed by the QA lead
5. Execute test cases and detect any bugs
6. Report bugs, if any and once fixed, run the failed tests again to re-verify the fixes