Linnéuniversitetet
Kalmar Växjö

Test plan

Linnéuniversitetet
Kalmar Växjö

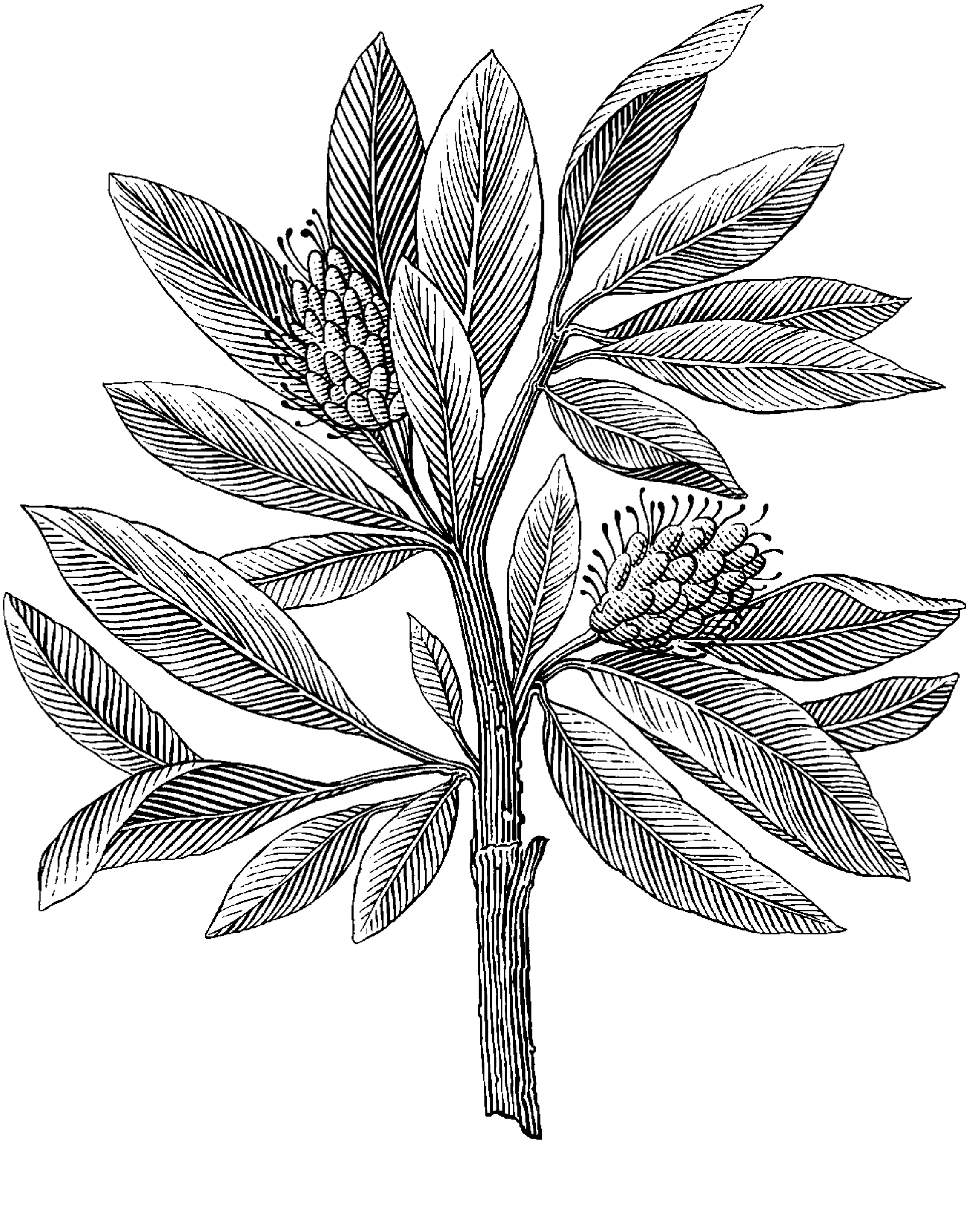

Table of Contents

Contents

1.Introduction

This document is about how test lead planned for testing the requirements for satisfying the stakeholders goals.

2.Requirements to be tested

This section consists of the followings:

1. All requirements which test in this iteration.
2. Level of testing for different requirement.
3. Technology used for testing different requirements.

Requirements	Level of testing	Technology	Priority	Type
1.1 use case 1 “Start Server” should be tested	Manual (System)	Exploratory&manual	High	Functional
1.2 use case 2 “Stop Server” should be tested	Manual (System)	Exploratory&manual	High	Functional
1.3 use case 3 “Request shared resource” should be tested	Manual (System)	Exploratory&manual	High	Functional
1.4 The source code should be released under GPL-2.0	System testing	Automated &manual	Medium	Non-Functional
1.5 The web server must follow minimum requirements for HTTP 1.1	System testing	Automated &manual	High	Non-Functional
1.6 The web server should be responsive under high load.	System testing	Manual	Low (only Windows10)	Non-Functional
1.7 The web server must work on Linux, Mac, Windows 10.	Manual	Exploratory	Medium	Non-Functional
1.8 The access log should be viewable from a text editor	Manual	Exploratory	Medium	Non-Functional
1 Web application needs to fulfil the requirements stated in the original requirement documen	Acceptance testing	Exploratory&Manual	High	Functional

2.1. Priority

Since provided use cases in legacy documents have the potential to be tested in many ways through manual tests, requirements 1.1,1.2,1.3 have the highest priority to test. The focus of this iteration is on manual testing.

3.Regression testing

After searching in source code, automation tests found which will run and check for coverage in testReport.pdf. SDC management wants a total code coverage of at least 85%. Any amount of less than 85% will result that the company should look for another application.

4.Requirements not to be tested in this iteration

Due to lack of time some requirements could not be tested in this iteration. Requirements 2,3,4,5, and 6 will test in next iteration.

5.Responsible persons

As already mentioned in testStrategy.pdf, there would be only 1 tester which is responsible for all tests and relevant documentation.

6.Risks and how to mitigate them

Whenever planning for something, there is always a chance to face risks, and the best way to reduce the impact of risks is to have a plan to encounter risks.

Risk	Possibility	impact	How to mitigate it
Having low knowledge about server and client programming	Medium	High	Start earlier and get familiar with socket programming
Due to pandemic, it would be easy to get sick and have less time	High	High	Best way is to start earlier and try to avoid any gathering or staying in public places with more than 10 persons.
Having some other tasks need to be done parallel to this task	Medium	Medium	Best way to reduce the impact is to try to have a pre-schedule plan and to follow it as much as possible.

7. Test planning

Since the test lead does not have enough knowledge about web programming, the test leas first step is to do exploratory testing to get familiar with MWS and socket programming. As the test lead mentioned before, this iteration is mainly planned for creating test cases for doing manual testing. For testing HTTP status codes, the test lead uses the Postman application. For that reason, the test lead needs to obtain enough knowledge of how to use it. After writing manual tests and running them, the result discusses in testReport.pdf. Finally, for doing performance testing, Apache JMeter will be used to test MWS under high load and see any changes in its performance.