Hayden Ackerman

Dr. Sean Hayes

CSCI 498 03

November 24, 2021

Test Plan Draft

# Introduction

* 1. Overview

This test plan describes the testing approach and overall framework that will drive the testing of the GUIMap program.

* 1. Goals

The goal of this test is to verify that the functionality of GUIMap works according to the specifications. The test will execute and verify test scripts, identify, fix, and retest all software bugs.

The final product of the test is:

* A production-ready software.
* A user-friendly UI.
* A comprehensive graph that displays data pulled from the software.
* An Ansible automation script.
  1. Constraints

This test will suffer from time-based constraints. Additionally, it will suffer from the developer being unfamiliar with some software components. The developer will have to consult documentation to overcome these constraints.

# References

* 1. Documents

This test utilizes two documents:

* Requirements.md & SeniorProjectProposal.md

# Test Items

* 1. Software

This test will utilize several different software products:

* SQLAlchemy 1.4.25
* Grafana Enterprise Edition 8.2.0
* SQLite 3.36.0
* Visual Studio Code 1.61.0

# Features to be Tested

* 1. Tested Features
* Nmap Scan Completion: All (applicable) flags of the nmap scanning engine will be used. Various networks will be scanned, including my home network, the university network, and potential virtual networks.
* Grafana Data Visualization: Scans will be visualized with the Grafana graphing engine. Testing will begin with small scans, listing very few results. By the end of testing, the goal is to get large-scale scans implemented within Grafana. The user will be able to save their data.
* Database Functionality: Scans will be saved to the database. Small-scale scan results will be the first to be implemented and then large-scale scanning results.
* UI Functionality: The UI will be tested on a Likert scale. The exact scale has not been nailed down, however a scale is being developed based upon user feedback.
* Ansible Automation Script: An Ansible Automation script will be configured to set up the ideal deployment environment. Common configurations are expected to be covered.

# Features Not to be Tested

* 1. Features not Tested

All features will be tested.

# Approach

* 1. Testing Methods
* Testing will consist of a case-by-case basis. For example, say that I start with the Nmap scanning function. If that passes, I move to the next testing module. I will keep testing until I run into a bug or fault. At that point, I will attempt to fix the bug/fault, and start back over. This process will be repeated until the product is capable of smoothly operating.
  1. Testing Levels

Component testing will be used to ensure that each individual piece of the software functions according to specifications. Most testing will be done via manual means, such as entering scan data and database data. This will ensure that the entered data is controlled. At a point in the testing, I will begin utilizing “unconventional” data, to test the robustness of the product. If automated testing is available, depending on usability, it may be integrated as well.

# Item Pass/Fail Criteria

* For the Nmap scanning, if the scanning completes without error, then it passes the test. A scan should at least indicate whether hosts are up or down,
* For the Grafana graphing engine, if the engine is able to display fine details of the scans, the test passes. A user should be able to narrow the results down by scan type, subnets, and so forth.
* For the database functionality, a user should be able to upload the scan results without receiving an error from the database itself. If the user is able to receive confirmation, both manual and automatic, from the database, the test passes.
* For the UI functionality, the UI will have to pass a Likert scale that will be discussed by the course instructor and the student. Additionally, all buttons and input fields will be functional. If all these conditions are met, the test passes.
* For the Ansible Automation Script, the configuration script will run without error on a test machine. If this succeeds, the test passes.

# Suspension Criteria and Resumption Requirements

* 1. Suspension Criteria

Testing will be suspended upon encountering a notable decrease in performance, a program-breaking bug, or segmentation faults. Another suspension criteria is if assigned test resources are not available when needed by the test team.

* 1. Resumption Criteria

If testing is suspended, resumption will only occur when the problem(s) that caused the suspension have been resolved. When a critical defect is the cause of the suspension, the fix must be verified by the testing team before testing is resumed.

# Test Deliverables

* 1. Test Plan

This document itself is considered a test plan.

* 1. Test Cases

No test cases have been created yet.

* 1. Test Scripts

No test scripts have been created yet.

* 1. Defect/Enhancement Logs

No Defect/Enhancement logs have been created yet.

* 1. Test Reports

No test reports have been created yet.

# 10. Test Environment

10.1 Hardware

Testing will be done on the following hardware specs:

* + - * AMD Ryzen 7 3700x CPU, 8c/16t @ 4.2GHz
      * Sapphire Pulse Radeon RX 5700, Driver Version 21.11.1
      * Corsair Vengeance RGB PRO 32GB (2x16GB) DDR4 3200MHz CL16 in Dual-Channel mode

10.2 Software

Testing will be done with the following software:

* Oracle Virtualbox Version 6.1.28 r147628 (Qt5.6.2)
* Ubuntu Linux, version 20.04.3 LTS
* Python 3.8.10
* Nmap version 7.80
* Microsoft Windows 10 Enterprise, 10.0.19043

10.3 Network

Testing will be done with the following network settings:

* Virtual NAT through Oracle Virtualbox

# 11. Estimate

11.1 Costs and Effort

Testing will not cost anything in a monetary sense. Testing will take at least 60+ hours of effort.

# 12. Schedule

12.1 Schedule

|  |  |  |
| --- | --- | --- |
| Task | Start Date | Completion Date (Estimated) |
| Create Project | August 28, 2020 | August 28, 2020 |
| Install Necessary Software/Libraries | August 28, 2020 | August 28, 2020 |
| Graphical User Interface (GUI) | August 28, 2020 | March 5, 2021 |
| Final Draft of Requirements | March 19, 2021 | April 6, 2021 |
| User Input | August 30, 2021 | September 14, 2021 |
| Create Database Schema | September 24, 2021 | October 10, 2021 |
| File Input/Output & File Conversion | October 16, 2021 | November 1, 2021 |
| Create a Test Plan | November 17, 2021 | December 5, 2021 |
| Testing Project for Bugs/Issues/Etc. | February 20, 2022 | March 2, 2022 |
| Compile Results & Analyze/Fix Bugs | March 2, 2022 | March 15, 2022 |
| Complete Project & Defend Project | April 15, 2022 | April 15, 2022 |

# 13. Staffing and Training Needs

13.1 Staffing and Training

This testing does not require any special staffing or training.

# 14. Responsibilities

14.1 Responsibilities

This testing requires two team members to fulfill specialized roles:

* Dr. Sean Hayes: Reviewing documentation, lead Quality Assurance engineer
* Hayden Ackerman: Lead developer, tester, and documenter

# 15. Risks

15.1 Risks

This test plan has some assumed risks:

* The project will not be completed in time
* Critical project-stopping bugs will happen
* Other schoolwork will take priority
* Data corruption leads to loss of the project

15.2 Mitigations

There are mitigations in place for the risks:

* Responsible time management will ensure that the project is completed in time
* Debugging will help narrow down problem pieces of code
* Balancing of schoolwork will ensure that equal attention is provided to the project
* Several backups will ensure that if any data corruption occurs, its effect will be minimal

# 16. Assumptions and Dependencies

16.1 Assumptions

This testing will have some assumptions:

* Constantly working on the project will not be possible
* Bugs will happen
* Deadlines will need to be extended

16.2 Dependencies

This testing has some dependencies:

* Scanning cannot be completed without the Nmap engine
* Storage of scanning results cannot be completed without the database
* The project cannot be completed without an IDE
* The project cannot be completed without Python
* Scanning cannot be completed without a network

# 17. Approvals

17.1 Approvals

The people that will have to approve the plan is Hayden Ackerman and Dr. Sean Hayes.