

Overall Test Plan

For the overall testing, we plan to do the testing on two different fronts. One dealing with the hardware and one dealing with the software. We will stress test the hardware as well as test to make sure that functionality is working correctly. Next, we will perform validation of different types on the webapp itself to ensure it's functionality works as expected with both security validation as well as basic functions working.

Test Case Descriptions

Hardware Test 1

This test will ensure that the hardware will work correctly under load.

This test will run simple stress tests so that we can test cooling and performance.

Inputs: Stress tests

Outputs: Cooling and CPU usage data

Normal

Blackbox

Performance

Integration

Results: Under simple stress testing, cooling remained at a steady level without performance suffering too much. There were also no issues with the Raspberry Pi crashing.

Hardware Test 2

This test will ensure that the hardware will work over a long period of time without rest.

This test will run the RaspberryPi under normal load so that we can test performance over long periods of time.

Input: normal load, time

Outputs: Cooling and CPU usage data

Normal

Blackbox

Performance

Integration

Results: Raspberry Pi was able to function over long periods of time without any issues in the logs.

Hardware Test 3

This test will test to see what happens when the hardware is put under multiple failure states.

The test will test what happens to the hardware after being denied power multiple times.

Inputs: Denial of power

Outputs: logs and hardware data

Abnormal

Blackbox

Functional

Integration

Results: Multiple failures did not seem to cause any large issues with the Raspberry Pi, however, we did not test things like static shock due to an electrical shortage due to possible destruction of the Raspberry Pi.

Web Server Test 1

This test will ensure that the webpage can be reached from multiple locations.

This test will be run from multiple places with internet connection aside from locally.

Inputs: Internet Connection

Outputs: Connection to the server

Normal

Whitebox

Functional

Unit

Results: The web page was able to be reached from outside local networks.

Web Server Test 2

This test will ensure that the webpage will function after hardware failure.

This test will be run by testing how the webserver reacts to a restart or failure in the RaspberryPi.

Inputs: Denial of power to RaspberryPi

Output: logs of webserver

Abnormal

Blackbox

Functional

Unit

Result: The webpage is able to function after hardware failure but requires a manual restart of the service since we did not want to set it up to run on boot due to design specifications.

Web Server Test 3

This test will test the ability for a user to login and authenticate themselves.

We test this through the user navigating through different pages and being able to login using the correct login information.

Inputs: Valid login credentials

Outputs: A successful login

Normal

Blackbox

Functional

Unit

Return: User was able to successfully validate themselves through credentials.

Web Server Test 4

This test will test the ability for a user to store their data in the database.

We will test this by creating new entries for the mining pool data through the use of the web app.

Inputs: mining pool credentials

Outputs: successful return of correct data from the database

Normal

Blackbox

Functional

Unit

Results: Users were able to store their mining pool addresses and receive a return of data from the API.

Web Server Test 5

This test will test the ability for a user to delete previously stored data in the database.

We will test this by using the web app to delete previously stored data and seeing if it is reflected in the backend database.

Inputs: User going through the deletion flow

Outputs: User data is deleted from the database

Normal

Blackbox

Functional

Unit

Results: A user is able to delete a mining address and have it be gone from their dashboard.

Web Server Test 6

This test will test the ability of a user to create an account.

We will test this by going through the userflow for a user to create an account.

Inputs: user credentials

Outputs: user account is created and stored in the backend database

Normal

Blackbox

Functional

Unit

Results: A user is able to successfully create an account through the use of the register page.

Web Server Test 7

This test will test the ability of a user to create an account using credentials that another user has already used.

We will test this by going through the user flow for a user to create an account, but with already previously used data in the database.

Inputs: Old user credentials

Outputs: User will be unable to create an account

Normal

Blackbox

Functional

Unit

Results: A user is not able to create an account with an email address that is already in use.

Test Case Matrix

	Normal/ Abnormal	Blackbox/ Whitebox	Functional/ Performance	Unit/ Integration
HW1	Normal	Black	Performance	Integration
HW2	Normal	Black	Performance	Integration
HW3	Abnormal	Black	Functional	Integration
WS1	Normal	White	Functional	Unit
WS2	Abnormal	Black	Functional	Unit
WS3	Normal	Black	Functional	Unit
WS4	Normal	Black	Functional	Unit
WS5	Normal	Black	Functional	Unit
WS6	Normal	Black	Functional	Unit
WS7	Normal	Black	Functional	Unit