**Test Plan Outline**

The suggested design for the system, build, and acceptance plan is used below. Any failures within the testing phase should be analyzed, corrected, and documented for future use.

1. **Introduction**
2. **Brief overview of the system -** Our IoT platform will be created within an Ubuntu Server. We will be using MySQL and LAMP stack to configure the IoT Platform. We will be using DigitalOcean to host to IoT platform.
3. **Document purpose and scope -** The purpose of the project is to create an IoT platform that will be able to scale both upwards and downwards. This is because as the website gets more users, we will need it to scale upwards so that it does not crash and when the website does not have a lot of users, we can scale it downwards to save money
4. **Test Procedures**
5. **Test Objectives –** The purpose is to test the system and fix any issues that may arise. The level of testing conducted during the implementation phase will have the application specialists reading the code and inspecting unit/module results.
6. **Testing Guideline**s – The Application specialists will create plans to test the functions of the system. The development team will create executables from the source code and test the executable according to the test plan. The management team, and the development team review the test results to ensure that all errors are identifies and corrected. The development will create user guides and system description documents. The application specialists will create the system test plan and in the independent acceptance test teal will use the plan during the acceptance test phase.
7. **Evaluation criteria** – The guidelines to determining if the test is successful/failure will be a test without any errors that cause major disturbance in the system. Any minor errors will be documented and corrected when members of the team have extra time. A successful test will include the system in operational stature. If the system is not working or is broken, the test should be considered a failure.

Table

Description automatically generated

1. **Error Correction –** Any system errors that are discovered should be fixed and retested to ensure that the issue is repaired. If the error occurs again, the team will continue to correct the error and keep rerunning the test until the error is corrected. Documentation of any errors will be made and filled accordingly. The librarian will update the configuration and test executables with the new revised corrections.
2. **Testing Summary**
3. **Environmental prerequisites -** Computer resources used needed to be recorded. External dates needed to be schedules and shared so that everyone is aware of the deadlines.
4. **Table summarizing the system**

|  |  |
| --- | --- |
| Unit Testing | Used to ensure that the unit can conduct the feature that it was design to do |
| Module Integration Testing | Developers will integrate individual units unto modules. The modules will be integrated in the system. |
| Build Testing | Used to ensure that the software is able to conduct the function that is needed by the system and to make sure that it’s the correct implementation of the design. |

1. **Test Descriptions (To be repeated for each test)**
2. **Test name** – Name will be needed to track the test
3. **Purpose of the Test** – Summary of the function that is being tested
4. **Method –** What steps are bring taken to test the function
5. **Expected results –** Expected results should be documented so that others can repeated the test and know what the end-goal for the test is
6. **Actual results -** What are the results that are received. This needs to be documented so that others can read the documents and repeat the test to see where the issue is and how they can fix it
7. **Regression Test Descriptions (Repeat items 4a to 4f for each regression test)**