# Test Plan

## Features to be Tested

All the product requirements must be tested, all requirements are outlined below.

| | Requirements | Type of Test | PASS or FAIL | CONTRIBUTOR |
|---|---|---|---|---|
| | **Functional** | | | |
| 1.1 | Users will be able to monitor temperature sensors in real time through a web browser client | | | |
| 1.1 | System will alert the users when the temperature exceeds the threshold | | | |
| 1.1 | System will automatically reconnect to sensors if connection is lost | | | |
| 1.1 | System will store past temperature data | | | |
| 1.1 | Users will be able to add or remove sensors | | | |
| | **Domain** | | | |
| 2.1 | The system will read and return values in degrees celsius. | | | |
| | **Product** | | | |
| 3.1 | System will be developed using python | | | |
| 3.1 | The readout of the sensors will be minimalistic, understandable and easy to read. (ex: Not returning machine code.) | | | |
| 3.1 | The sensors will publish their readings at a set interval and not lag or lead this timed interval. In this case, every 500ms should suffice. | | | |
| 3.1 | The system will restart automatically if it goes down for any reason (ex: power outage) | | | |
| | **Organizational** | | | |
| 4.1 | System will implement automated testing | | | |

| | | | | |
|---|---|---|---|---|
| 4.1 | System will implement the publish-subscribe design pattern | | | |
| 4.1 | System will incorporate continuous integration and deployment | | | |
| 4.1 | System will implement the chain of responsibility design pattern | | | |
| 4.1 | System will implement the adapter design pattern | | | |
| 4.1 | System will use MQTT | | | |
| 4.1 | The system will be able to easily integrate a new sensor if required. | | | |
| | **External** | | | |
| 5.1 | The system must be secure, therefore a password will be required to access the data and administrative commands. | | | |

## Type of tests

The types of tests used for this project are listed below.

- Unit test
- APIi testing
- Integration test
- System test

## Risk and Issues

| Risk | Mitigation |
|---|---|
| Member lacks skill or knowledge for testing | Ensure that all team members understand and know how each requirement Is implemented. If any member is unsure they must communicate with other members so that they may teach or take over the test. |
| Schedule is not possible to follow and parts are not completed on time. | The requirements will be prioritised so that the most important and critical ones are completed first. |

| | |
|---|---|
| Requirements are not implemented properly and repeatedly fail tests. | Member working on the requirement should enlist the help of other members to fix the software. |

## Who will test?

All members of the team will test requirements, ideally a member different than the member who implemented the requirement will do the test. This so that any requirement gets looked at by multiple team members.

## When will test occur?

The tests will occur as soon as the software for each requirement is completed, and there is a team member available to test. If the software is tested as it is produced integration at the end will be easier since we know the previous components work or do not.

## Test Objective

The purpose of the tests are to verify that all components of the system work as intended, and that the system as a whole works. The tests should focus on the systems that monitor and analyse the data from the temperature sensors, and ensure that they are implemented using the proper techniques.

# TEST CRITERIA

Suspension Criteria (number of tests to suspend testing and fix cases): 40%

Exit Criteria, specifies the criteria that denote a successful completion of a test phase

- Run rate: 100%
- Pass rate: 80%