FISHMO
Semi-Automated Aquarium Monitoring System

Software Test Plan
Revision 2

April 9, 2015

Group 5

Rhett Amin - 1060085
Brandon Da Silva - 1057434
Dan Esteves - 1152328
Michael Liut - 1132938

Prepared for:
Computer Science 4ZP6 Capstone Project
Instructor: Dr. Rong Zheng
Fall/Winter 2014-2015

Contents

1 Introduction

This document is the Test Plan for FISHMO SAAMS. This document will address the various items and elements and how they will be tested, within the nature of the system. Furthermore, it will cover everything from testing the inputs from the sensors connected to the Raspberry Pi, to the controls on the web application.

The primary focus of this plan is to ensure a reliable, user-friendly, and safe software-hardware integration that will ensure the user can effectively take care of their aquatic system.

This project will use the following testing methods: unit testing, system testing, acceptance testing, and usability testing as well as other testing approaches. The details to these methods and others are detailed further in this document.

1.1 Objectives

This project will be developed in accordance to the guidelines specified by the Agile Development principles. This document currently contains the most up-to-date information on the system and its features. As the system continues to develop, it will be pertinent to review and keep this up to date. Furthermore, software testing will occur after every update to ensure proper installation of all firmware and the full functionality of the device's peripherals.

This test plan's main objective is to ensure that the system is reliable, user-friendly, and consistent. This will be done by testing as the system is being developed at each major milestone. This test plan will ensure that the system to be delivered meets the users expectations of it, and will ensure that the user can use the system easily. The system is split into two sections, which are: the Raspberry PI, and the web application interface. Major Milestones for these parts include: connecting all sensors and ensuring that they work properly, Creating a working prototype of the interface, connecting both the hardware and the software together, testing each function of the interface.

1.2 Testing Strategy

1.2.1 Component (Unit) Testing

This test method will ensure that each element of the FISHMO system works as intended.
Items to Test: Input from each sensor, adding/removing fish to the database, proper population of database, proper updating of database variables, software outputs to the hardware(light, heating).
Approach: For the hardware-software communication elements, we will test this by by using a terminal interface, then later on the web app interface. To test the other elements, we will test using the web app, and confirming that each element works as intended.
Pass/Fail Criteria: Each element fails if it does not meet the required functionality as it is intended to. It passes if it does (i.e. proper population to the database - FAIL, because it only populated half the database).
Roles: Each member of this project will test a certain element of the system.
Schedule: Every time a major milestone is reached we will perform a unit test on the new element added.

1.2.2 Integration (System) Testing

This test method will ensure that design objectives are met and ensures that the software, as a complete entity, complies with operational requirements.
Items to Test: The entire FISHMO system and components
Approach: This will be tested as if we are the user using this product. We will use every function and element of the system to ensure that it works as intended.
Pass/Fail Criteria: This test will pass, if the system meets every requirement. It fails it it doesn't meet any of the required functionalities.
Roles: One member will test the system.
Schedule: This test will be done at the very end, to ensure the system is complete.

1.2.3 Acceptance Testing

This test method will be conducted to determine whether or not a system satisfies the acceptance criteria and to enable the customer to determine whether or not to accept the system.
Items to Test: If the user accepts the system as a viable tool to take care of their fish.
Approach: This will either conducted using an user that is not us, to ensure an actual customer would use this.

Pass/Fail Criteria: Fail: If the user does not accept the system; Pass: If the user accepts the system
Roles: Each member will look for a third party user to test the system.
Schedule: This will occur once the system is completed.

1.2.4 Interface Testing

This test method will ensure that the system properly translates data between the hardware and the software.
Items to Test: The proper interaction between the hardware inputs and the software. Also the software output to the relevant hardware controls.
Approach: This will be done when the web app is in its prototype stage, and when the hardware is setup with all its sensors ready to begin data transfer to the software.
Pass/Fail Criteria: Fail: If the system fails to transfer data, or transfers incorrectly; Pass: If the system transfers data correctly
Roles: One member will test if the interface between the software and hardware is completed.
Schedule: This will occur when the software and hardware are ready to be connected.

1.3 Scope

The software test plan will be updated at each major milestone, and will include updates to the testing methods, and any addition/subtraction as necessary. This test plan will be contained on our Github page which can be accessed by only the members of this project, and the TA for the course. Updates about the test plan will be conducted on "Slack" a third party software application used for every conversation we have regarding the project. The changes discussed will then be reflected in a new test plan revision.

1.4 Definitions, Acronyms, and Abbreviations

Term	Definition/Acronym
FISHMO SAAMS	FISHMO Semi-Automated Aquarium Monitoring System
Pi	The Raspberry Pi is a credit-card sized computer that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games.
User	A person who interacts with the FISHMO system
NOOBS	NOOBS (New Out Of the Box Software) is an easy operating system install manager for the Raspberry Pi.
GUI	Graphical User Interface
Camera Board	The Camera Board is a small PCB that connects to the CSI-2 camera port on the Raspberry Pi using a short ribbon cable. It provides connectivity for a camera capable of capturing still images or video recordings.
3D Printer	A process for making a physical object from a three-dimensional digital model, typically by laying down many successive thin layers of a material.

1.5 Revision History

Revision #	Revision Date	Description of Change	Author
0	October 22, 2014	Initial Test Plan	All members
1	January 13, 2015	Edited Test Plan integration section	All members
2	April 9, 2015	Updated Test Plan	All members

2 Test Items

2.1 Program Modules

Once a feature has been built, sensing the water temperature for instance, the robustness of the code must be tested. This entails testing the code for common errors such as syntax as well as testing the code to ensure that it functions properly. The features are intertwined and dependent on other components so as a result this must be tested against that to ensure proper functionality. We can however test certain components from the hardware through the terminal, but would require the web-page to final correct testing. The inputs, processing and outputs of each module will be tested to confirm it is performing correctly.

Not only is the software supposed to work properly on its own, it should be able to work coherently with the hardware. This testing must be done to ensure that the expected action from the hardware is actually performed.

2.2 User Procedures

After something has been refined or modified whether it be software or hardware, the user documentation that supports the area of changes must be tested to ensure that the documentation is consistent. If there are feature changes or updates, the user documentation must reflect these changes.

2.3 Operator Procedures

In order to test for a fully operational desktop application, we will test the application in a controlled environment rigorously. We will test the application by not only testing for functionality, but also performance as it will endure the most strenuous of conditions. Once the desktop application is fully functional on one platform, it will then be tested for consistency and credibility on other browsers. Overall the web application will be the main focus of testing and debugging. The main purpose of testing the web application will be to try to force and issue that the team can fix and foolproof before releasing to the public.

3 Features To Be Tested

The following features will be tested in the system:

3.1 Adding Fish (Populating the Database):

Prior to release, our database will be verified with an expert in the fish industry to ensure optimal conditions for the fish's survival in our aquatic environment. Once completed, we will attempt to add fish to the database while working under controlled conditions; then the device will be released to the public for beta testing.

3.2 Modifying the Database Variables:

Prior to release, our database will be verified with an expert in the fish industry to ensure optimal conditions for the fish's survival in our aquatic environment. Once completed, we will modify, add or remove to remove variables from the database while working under controlled conditions; then the device will be released to the public for beta testing.

3.3 Accessing the Camera:

Prior to release, we must test to make sure the the user has easy and live access to the camera stream without bugs or interruption. Once we can verify that the camera stream is live and accessible via the internet we will move forward.

3.4 Accessing the Temperature Sensor:

Prior to release, the temperature sensor will be tested to verify that the correct readings and measurements are being interpreted through the PI and over to and through the user interface. Once verifying that the readings are correct, and that the script is error and bug free we will release the sensor to be tested in extreme conditions

3.5 Accessing the Heater:

Prior to releasing the FISHMO application to the public we are responsible for testing the heating system, we will test the heating system to guarantee that it provides the ability to increase the temperature of the water given signal through the PI using Python and the web application.

3.6 Accessing the Water Flow Sensor:

Prior to release, the water flow sensor will be tested to verify that the correct readings and measurements are being interpreted through the PI and over to and through the user interface. Once verifying that the readings are correct, and that the script is error and bug free we will release the sensor to be tested in extreme conditions

3.7 Accessing the LEDs:

Prior to release, the last thing that we will test for confirmation of use will be the LEDs. We will ensure that the LEDs are accessible through the user interface and that the desired outcome of the action is performed.

4 Features Not To Be Tested

4.1 Power Supply:

As FISHMO SAAMS does not come with a battery or power supply of any kind, it requires AC power to operate. Thus we cannot guarantee the reliability of the power supply nor need to conduct testing on it, as it is assumed to be reliable.

4.2 The Tank:

As FISHMO SAAMS can be placed in any aquatic system, the durability, reliability and overall structural integrity of the tank is outside of the purview of our device and therefore assumed to be reliable and structurally sound.

4.3 Water Level (In Tank):

As each aquatic system will vary, and it is outside the purview of FISHMO SAAMS, the tanks water level will deviate per customer and therefore is assumed to be filed to the optimal level.

5 Approach

The types of testing being performed on FISHMO SAAMS for robustness include, Component (Unit) testing, Integration (System) testing, Interface testing, Security testing, Recovery testing, Performance testing, Acceptance testing and Beta testing.

5.1 Component (Unit) Testing

Both component (hardware) and unit (software) testing are to be performed in this phase. Our components include: a temperature sensor, a water flow sensor, a water pump, a RGB light strip, a camera, a network adapter, a water heater, a breadboard, wires, and the PI. All of which are rigorously stress tested prior to deployment. Furthermore, our code must meet certain requirements prior to execution. Once passed, during the initial stage of compilation (when the system is turned on), the system will become active and ready for use.

5.2 Integration (System) Testing

This is an important process to ensure harmony between our software and hardware components. As FISHMO SAAMS is here to ensure the most optimal environment for the fish living in the aquatic system, we must ensure our integration is flawless, robust and that the transition for the end user is seamless. Once again, this is done through component/unit and system stress testing.

5.3 Interface Testing

Interface testing ensures the ease of usability of our product. Keeping our overall interface clean and simple goes a long way for the user's ability to learn the functionality of our software. Furthermore, unit and beta testing plays a huge factor into this component as the best interface testers are the end users. Interface testing is conducted during the unit testing phase and beta testing phase of our software development cycle.

5.4 Security Testing

This section is to test the robustness of both the design and implementation of our software to protect the data and resources contained in and controlled by it. This directly applies to our database, as we require ample security for users personal account information and login credentials. Data must be encrypted using industry standard ciphers. Testing of this phase would most-likely occur by a third-party during the beta testing phase.

5.5 Recovery Testing

Recovery testing ensures that the user's settings will be backed up locally on the PI (OEM software will be partitioned into a recovery section) and that the user's personal data and the optimal settings stored on the database are backed up accordingly. All backups must be encrypted using industry standard ciphers and verification of these backups are of utmost importance. During this process we verify backups have been conducted, otherwise the necessary precautions are taken to avoid data loss.

5.6 Performance Testing

Performance Testing is key in ensuring that our user experience is a good one. Here we test for the response time of sensors, activation of components and measure the time is takes for changes to provide feedback to the end user.

5.7 Acceptance Testing

Acceptance testing occurs in the unit testing phase. As the software integration is required to pass the unit tests constraints within machine precision accuracy. These requirements will be set based on optimal feedback times.

5.8 Beta Testing

Beta testing is one of the most important testing factors when developing a product, as the end users feedback will allow FISHMO to succeed in the future. This process will allow the user to test the integration of the hardware and software, but more so the robustness of the overall system and the usability of the user interface. We will allow for feedback to be provided to us from the end user via website forms or direct email responses.

6 Pass/Fail Criteria

Suspension, Resumption and Approval Criteria are all used to determine whether each item has passed or failed this step in the testing cycle.

6.1 Suspension Criteria

These are conditions that need to occur, to suspend testing:

- The database is not properly populated with relevant data when the user adds a fish to the database
- The database entries are not removed successfully when the user chooses to remove a fish from the database
- The database is not successfully updated when the user saves changes made to it
- The user cannot make changes to the database
- The camera cannot be accessed, or does not display properly
- The system cannot access the temperature sensor, the heater, and/or the water flow sensor
- The LED light cannot be accessed and/or state cannot be modified

6.2 Resumption Criteria

These are conditions that need to occur, to resume testing:

- The database is properly populated with relevant data when the user adds a fish to the database
- The database entries are removed when the user chooses to remove a fish from the database
- The database is successfully update when the user saves changes made to it
- The camera is accessed properly, and the live feed is displayed Proper access to the temperature sensor, the heater, and the water flow sensors exist
- The LED light can be accessed and state can be modified

6.3 Approval Criteria

The formal testing approval process involves each member of this project rigorously testing each functionality of the system, as to ensure that each success was not a fluke, and that the results are reliable. Each member will test each condition separately. The following are conditions that need to occur, to be approved as completed:

- The database is properly populated with relevant data when the user adds a fish to the database
- The database entries are removed when the user chooses to remove a fish from the database
- The database is successfully update when the user saves changes made to it
- The camera is accessed properly, and the live feed is displayed Proper access of the temperature sensor, the heater, and the water flow sensor
- The LED light can be accessed and state can be modified

7 Testing Process

7.1 Test Deliverables

7.1.1 Component (Unit) Testing

This test is to ensure each component and function of the system works as intended.
Input: Reading the sensors, sending a signal to the heating pad to turn it on/off, adding fish to the database, removing fish from the database, modifying variables in the database, accessing the camera stream.
Output: Providing feedback to the user via system widgets when abnormal sensor readings occur, the heating pad state changes, the database is populated with new information when the fish are added, the database updates with the removal of fish, the database updates when variables are modified, and when the camera is being streamed.

7.1.2 Integration (System) Testing

This method is used to test if the system works together as a whole.
Input: Reading the sensors, sending a signal to the heating pad to turn it on/off, adding fish to the database, removing fish from the database, modifying variables in the database, accessing the camera stream.
Output: Hardware and software components interact as expected during the unit testing phase (located above).

7.1.3 Interface Testing

This test method is used to ensure components pass controls and data correctly between one another.
Input: Sensors (hardware components) send specific data to the software, the software then interprets that data and sends the signal to the necessary location.
Output: The heating pad is turned on when the temperature sensor detects the overall water temperature to be too cold (set by user) or the camera is turned on when the user activates the control.

7.1.4 Security Testing

This test method checks to see if the database can be accessed by someone else, protecting our user's data and to ensure the integrity of our application.
Input: Attempting to access another person's account, trying to access the FISHMO PI controller, trying to change another user's local information by breaking our ciphers.
Output: The system is not penetrated by attacks of any kind and user data is protected.

7.1.5 Recovery Testing

This test method is used to ensure that the database contents are saved correctly and that any OEM software is partitioned for recovery purposes.
Input: Modifying the user's local or database (e.g. add a couple of fish to the database) and then restart the system.
Output: The system should maintain modified information, and restore the local data with the correct database data.

7.1.6 Performance Testing

This test method is to verify that the user's software experience transitions smoothly with no latency issues and minimalist response times.
Input: Using the software, find ways to stress the system (e.g. adding 100 fish to the database, switching between modes quickly, or even turning it on/off).
Output: The system should behave smoothly and respond quickly to user actions.

7.1.7 Acceptance Testing

This test method checks to see if the user accepts the system as a better method of taking care of their fish.
Input: Beta testing, the user must own and operate a FISHMO SAAMS daily.
Output: The user accepts or rejects the system.

7.1.8 Beta Testing

This method is used to ensure the way we are developing the system meets the functional requirements, and to detect any failures or faults missed during the other phases of testing.
Input: User feedback on the product.
Output: User finds the system acceptable, or finds faults and reports them to us.

7.2 Testing Tasks

7.2.1 Component (Unit) Testing

Each component needs to be in a usable/complete state for testing. After initial testing, prior to integration, the components are independently tested.

7.2.2 Integration (System) Testing

The system needs to be put together, both hardware and software.

7.2.3 Interface Testing

The software and hardware components need to be fully functioning, and completely integrated to begin testing their communication.

7.2.4 Security Testing

The database and website application should both be completed before this stage of testing begins.

7.2.5 Recovery Testing

The system should be completed before this testing begins.

7.2.6 Performance Testing

The system should be completed before this testing begins.

7.2.7 Acceptance Testing

External users needs to be found and willing to test the system. The system be completed before this testing begins.

7.2.8 Beta Testing

External users needs to be found and willing to test the system. The system be completed before this testing begins.

7.3 Responsibilities

Every stage of testing must be completed fully and thoroughly, verified by every group member, ensuring the utmost accuracy in our results.

7.4 Schedule

The system will be tested twice, once when it is in a working state, and once when the system is in its fully functional form (i.e. the final product).

8 Environmental Requirements

8.1 Hardware

The computer hardware and network requirements needed to complete test activities include but are not limited to:

- Raspberry Pi
- Temperature Sensor
- Breadboards + wires
- Water flow sensor
- Water pump
- Water heater
- Network adapter
- Web-camera
- RGB Light Strip

In order to test the materials above we will need to simulate a controlled environment of an aquatic animal living in comfortable conditions as we test the hardware with a volt meter.

8.2 Software

Prior to public release, our software undergoes the most rigorous of unit testing, ensuring flawless functionality. This is one of the utmost important tasks our software must pass. Specifically, our software has three components to it. First, our Python code implemented directly onto the PI, this is used to manipulate the PI as a controller and it's components. Secondly, we use SQL and PHP for the back-end scripting of our database. PHP is also used to integrate the database with the last component, our web application written in HTML with CSS. With all three of these components on a unified front allow for the seamless transition and optimal user experience of FISHMO SAAMS.

8.3 Security

The user's data information for their FISHMO SAAMS is stored locally on the physical device. Every time the user wants to access their readings, it calls the Pi for readings once the web app is refreshed. Until that point in time, all of the data is stored locally on the Pi until it is requested. With the information being collected only when requested can keep the information from being tampered if it we are sent to the user periodically.

8.4 Tools

Tools used in aiding the production and testing of the FISHMO product includes: A laptop(s) or Desktop used as a main base of operating the Raspberry PI and creating a web app using MySQL, PHP, Bash and Python. Apart from that other tools include a hard drive which stored and was used for live access to and from the database. A Voltmeter was used to test currents in and around the circuits wired to provide accessibility and access from the general input/output pins to the raspberry pi. Lastly the with the aid of resistors and transformers we are able to supply proper amounts of power to our outputs such as the RGB strip, sensors etc.

8.5 Risks and Assumptions

External risks and assumptions leave huge room for impact on a project. Risks and assumptions have the ability to interrupt, break and crash any operation on FISHMO. For this reason is why users have to be aware of external risks such as we are when testing in a controlled environment. Risks include: Aquatic life or human life interfering with wiring of any sort, power shortages when running over-sized heaters and fans without correct transformers and damage(s) on the aquarium or other hardware or software.

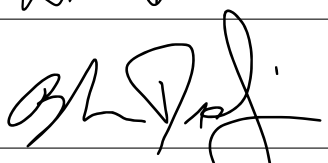
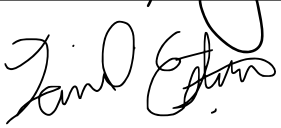
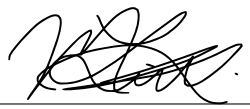
9 Change Management Procedures

This Test Plan will be kept updated on an ongoing basis in the project Github repository, located at: `https://github.com/MichaelLiut/FISHMO`

The content in this Test Plan is subject to change as the project progresses. Changes to the Test Plan will be written then approved by the group members of this project on a change by change basis as needed. All members of the project must be e-mailed, instant messaged or told in person about required changes, but no formal review is strictly required.

10 Plan Approvals

Approved by:

Name	**Student #**	**Signature**	**Date**
Rhett Amin	1060085		April 9, 2015
Brandon Da Silva	1057434		April 9, 2015
Daniel Esteves	1152328		April 9, 2015
Michael Liut	1132938		April 9, 2015