California Polytechnic State University, San Luis Obispo

Engineering-461

Multidisciplinary Senior Project

Pier2Pier: Final Report

Authors:
Kevin Nelson
Tony Miller
Paul Maalouf
Chahan Shah

Advisor:
Dr. John Ridgely

Sponsor:
Tom Moylan

June 7, 2015

```
______ _          _____ ______ _
| ___ (_)        / __  \| ___ (_)
| |_/ /_  ___ _ __`' / /'| |_/ /_  ___ _ __
|  __/| |/ _ \ '__| / /  |  __/| |/ _ \ '__|
| |   | |  __/ |  ./ /___| |   | |  __/ |
\_|   |_|\___|_|  \_____/\_|   |_|\___|_|

 ><  ______         ><              ______
    | ____ |   ><     ******       | ____ |
    ||    ||        **********     ||    ||
~~~~||    ||~~~~~~~~~~~~~~~~~~~~~~||    ||~~~
~~~~||~~~~||~~~~~~~~~~~~~~~~~~~~~~||~~~~||~~~
```

Statement of Disclaimer

Since this project is a result of a class assignment, it has been graded and accepted as fulfillment of the course requirements. Acceptance does not imply technical accuracy or reliability. Any use of information in this report is done at the risk of the user. These risks may include catastrophic failure of the device or infringement of patent or copyright laws. California Polytechnic State University at San Luis Obispo and its staff cannot be held liable for any use or misuse of the project.

Table of Contents

List of Figures

List of Tables

1 Executive Summary

The current project team made several major advances taking the system from being a collection of disconnected pieces of hardware to a networkable collection of devices with the groundwork laid for a more secure, rubust, and user-friendly platform for future groups to build upon. The project consists of a camera system contained within a watertight pod which is moved vertically along a fifteen meter track attached to a pier piling. Previous project teams implemented the track, pod electronics, and the pod itself. The current team developed a web page, a means for video streaming, a backend server, and an interface for servo control.

The current team developed a web page with live high definition video streaming and an interface for the user to control the vertical position of the pod. The user interface handles interactions by sending asynchronous JavaScript.

The video streaming was implemented by encapsulating compressed video in Flash and serving it via a crtmp server.

The backend server was written using Ruby on Rails; it serves html to clients and handles ajax requests.

The servo control system was programmed as a state machine that changes states in response to commands sent from the backend server.

The fact that this project has been a continuation of prior groups' work is reflected in the design process, which has been more a means of approach towards consolidation than creation. The approach taken with the constraints given has been to utilize the tools best suited to the current project and to lay the groundwork for further advancement.

2 Introduction

2.1 Sponsor background and needs

The sponsor for this project is Tom Moylan with the Cal Poly Coastal Marine Sciences Department. Tom Moylan is the manager of the Cal Poly Pier, a half-mile long pier located in Avila Beach.[11] Mr. Moylan requested a means by which to view underwater life in real time at the pier that is accessible via the Internet across the world. This accessibility is to extend to researchers, students, and elementary school teachers. The system is meant to move the camera up and down along a pier piling at the control of the viewer.

2.2 Formal problem definition

The goal for this project is to develop a remotely controllable means by which to view underwater life at the Cal Poly Pier. The system is to be accessible over the Internet. It is specifically geared towards researchers and elementary school teachers who want to view underwater life in real time. It is meant to be controllable by the viewer, so that the height and angle of the mechanism can be adjusted. Multiple people shall be able to view the output of the system in real time, while only one person shall have the capability to control the device at any one time. Access to the system will be limited based on the type of user. Researchers shall have the greatest level of access while using the system to complete research activities. Elementary school teachers will have a more limited level of access to the system when it is not in use by a researcher. The general public will have access to the system when it is not in use by researchers or elementary school teachers and the system is not under maintenance.

It is difficult to experience underwater environments for marine biologists and other researchers. They must be located near the ocean and have the funds and training to use Scuba diving equipment. Underwater conditions may prove unacceptable for accurate data recording or not be clear enough for close enough inspection of the environment or object of interest.

Underwater life is vastly different from life on land. Many children live hundreds of miles away from an ocean and may have never seen one in person. Yet the ocean contains most of the earth's life and is important for children to understand[44]. Being able to see underwater life in real time from across the world would be invaluable for educators trying to help their students understand the complexities of underwater life.

The utility of this project is justified by the lack of a comparable existing system and the unique challenges that must be faced in completing the implementation. From the background research, it is clear that there is currently no drop-in solution to this unique set of engineering challenges. The underwater aspect of the project constitutes the majority of the challenges to be faced and is likely the reason that live underwater viewing systems are not as widely available as those above ground.

3 Background

This project is a continuation of several years of prior work by other students. The unique demands of this project have not been addressed in any other educational or commercial product. Several groups do purport to live-stream underwater camera footage, though they are not remotely controllable. They seem to present a fixed camera position that would require a specialized diver to move or re-position[18].

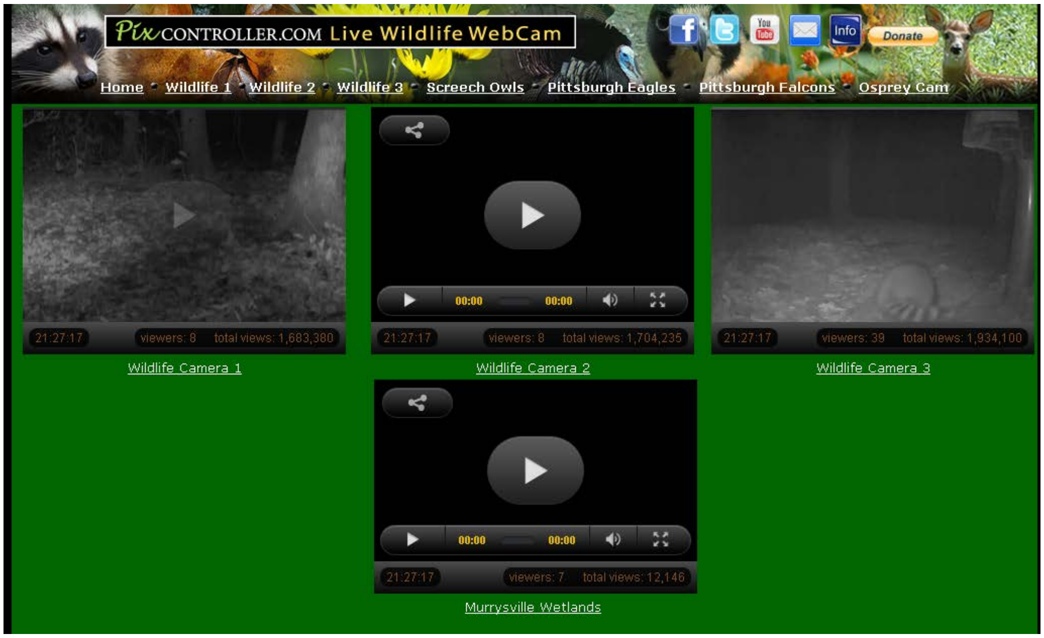


Figure 1: Multiple simultaneous views.

One alternative to giving the user direct control of the camera is to simply present multiple camera views on the same page. This way the user has the perception of being able to look around by physically looking at different camera feeds like on pixcontroller.com[30], shown in Figure 1. This has advantages and disadvantages compared to the current project. One advantage is there are no moving mechanical parts on site, which means there is much less required maintenance and fewer points of failure. A disadvantage of this system is the user does not feel immersed in the scene and has no control over their view. This could be troublesome for researchers who need to access several specific views repeatedly. Also, the general public will feel much less engaged in the experience if they cannot see their actions having an impact on the scene.This system also differs from the current project in that it does not have underwater cameras, only land based ones. This means the system is much easier to maintain and expand as necessary, with no diving or motor system involved.

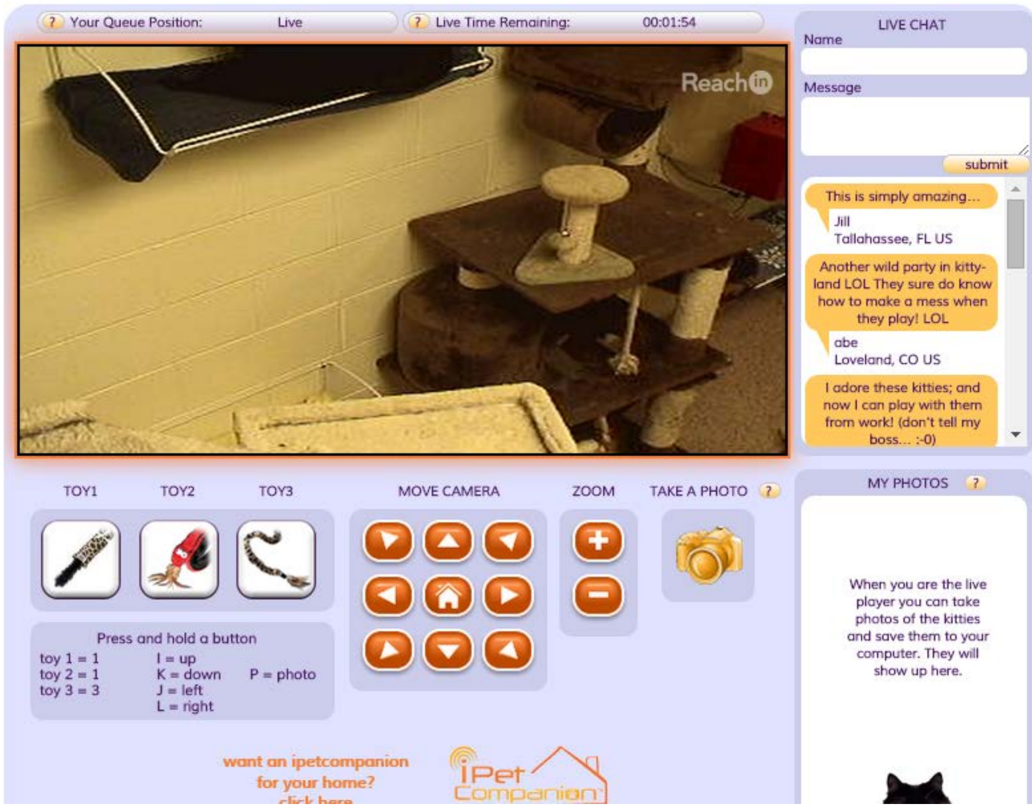


Figure 2: User controllable web camera.

There are user controllable web cameras that can be accessed via the web; however, most are slow, poorly maintained, or do not work. These sites also only allow the user to pan, tilt and zoom the camera; they have no means of actually moving the location of the camera. Our design will be able to change the depth of the camera and give the user more control. Figure 2 shows an examples of such a web cam that is more capable than most found online.

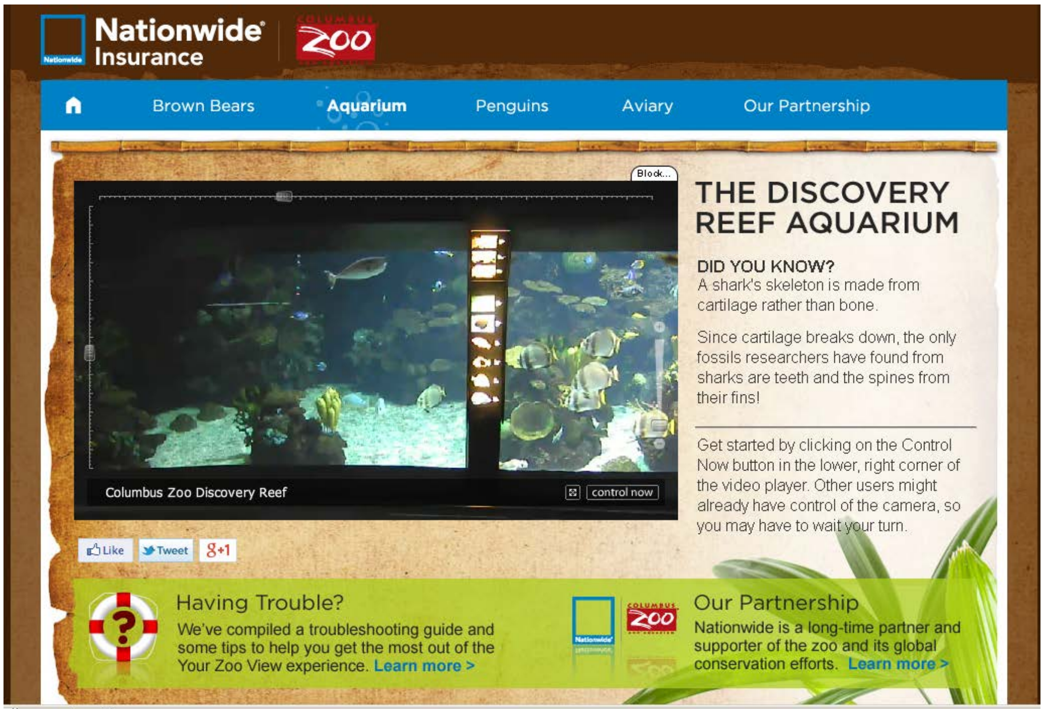


Figure 3: User Controllable aquarium camera.

The Columbus Zoo has a controllable web camera directed at an aquarium display online[15]. This camera gives the user a controllable camera that can be zoomed, panned, and tilted, shown in Figure 3. However, the actual camera is mounted to a wall outside of the water tank. Therefore this team did not have to face the difficulties of having an electronic system in direct contact with water. The interface is fairly easy to use and has adequate response time. However, while testing this system reliability was quite poor. After just a few minutes of use a “Your connection to the camera has timed out” message appears. Since the camera is located so far from the tank it breaks the feeling of immersion a system with a water based camera would offer.

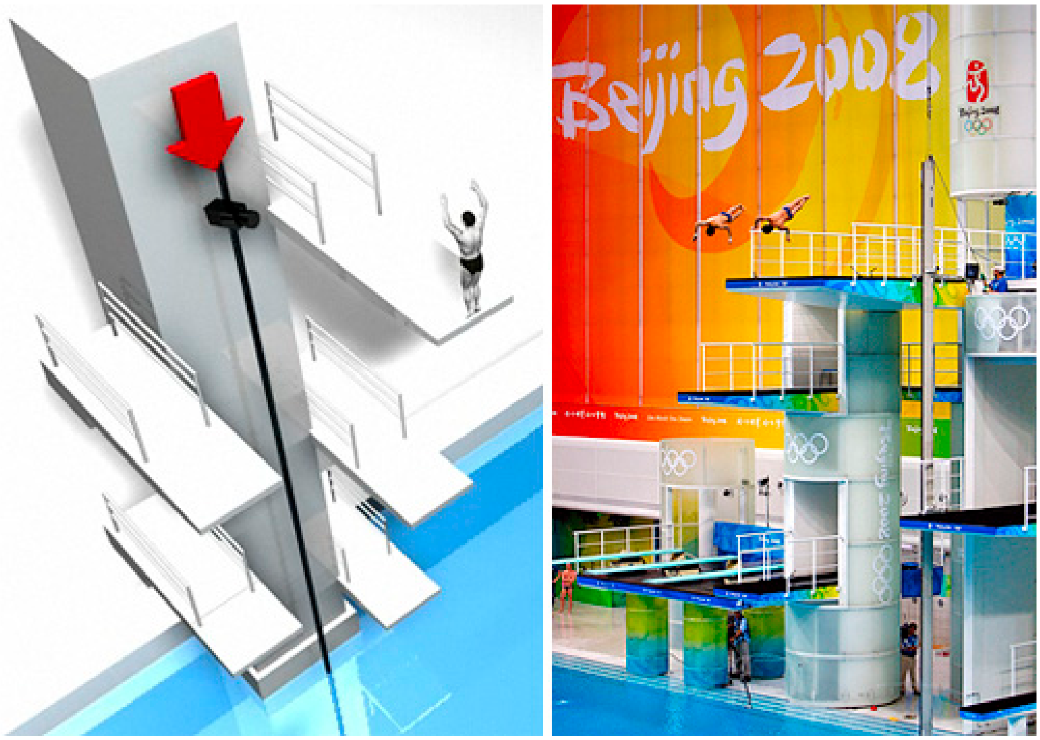


Figure 4: Olympic Dive Camera.

In terms of camera movement, there is an Olympic dive camera, Figure 4, that allows a camera to move vertically in order to track divers in their descent. This device drops the camera at free fall speed as the diver falls, and then reels it back to its initial position for the next dive[43]. This differs from our design as our device requires a controlled ascent and descent to fixed locations along the pier piling.

Figure 5: CENTR 360° video.

There are a few approaches for allowing the viewer to pan the camera while it is in use. There can be a mechanical system that changes where the camera is pointing. Another option would be to have multiple cameras in the pod, that point in varying directions and allowing the user to switch feeds between cameras. There is a camera built by a company called CENTR that records 360° video[12], shown in Figure 5. This device would enable the user to see what is going on all around the pod at the same time and will enable them to be aware of happenings about the pod that aren't necessarily in their view. A drawback of such a device is that it would use much more bandwidth than a typical camera. The pier has a low speed Internet connection that will struggle with streaming a single video feed at a time. Bandwidth limitations on the pier may make a CENTR type camera impractical for the time being.

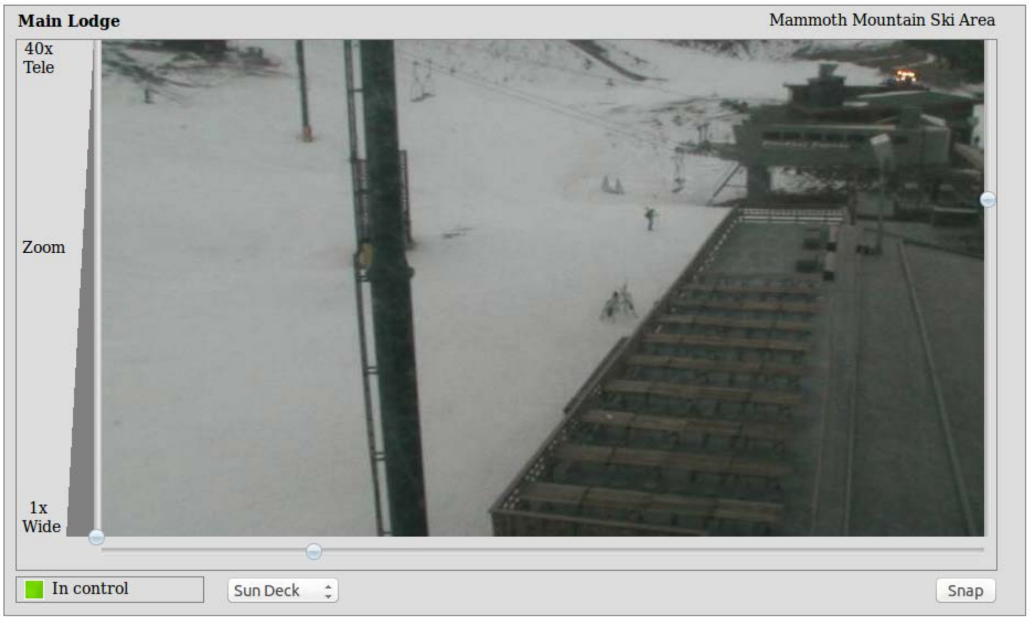


Figure 6: Main Lodge Cam at Mammoth Mountain.

The Mammoth Mountain web cam[25], shown in Figure 6 has many similarities to the current project. The Mammoth web cam allows the user to view a scene in an area with rather extreme weather conditions.It has user controls that are accessible to one viewer at a time. Each user is limited to a set amount of time when other users have requested access. It is different from the current project in that it does not allow users to schedule control access ahead of time and uses a simple first come, first serve queue to grant access. It is also different from the current project in that instead of controlling position and horizontal rotation like the current project the Mammoth web cam gives the user control of horizontal rotation and camera zoom level. Like the current project the Mammoth web cam has a saved location feature, however, that implementation is different from the current implementation in that the user cannot create custom saved locations. These locations are also not user specific and are accessible to all site visitors, unlike with the current project.

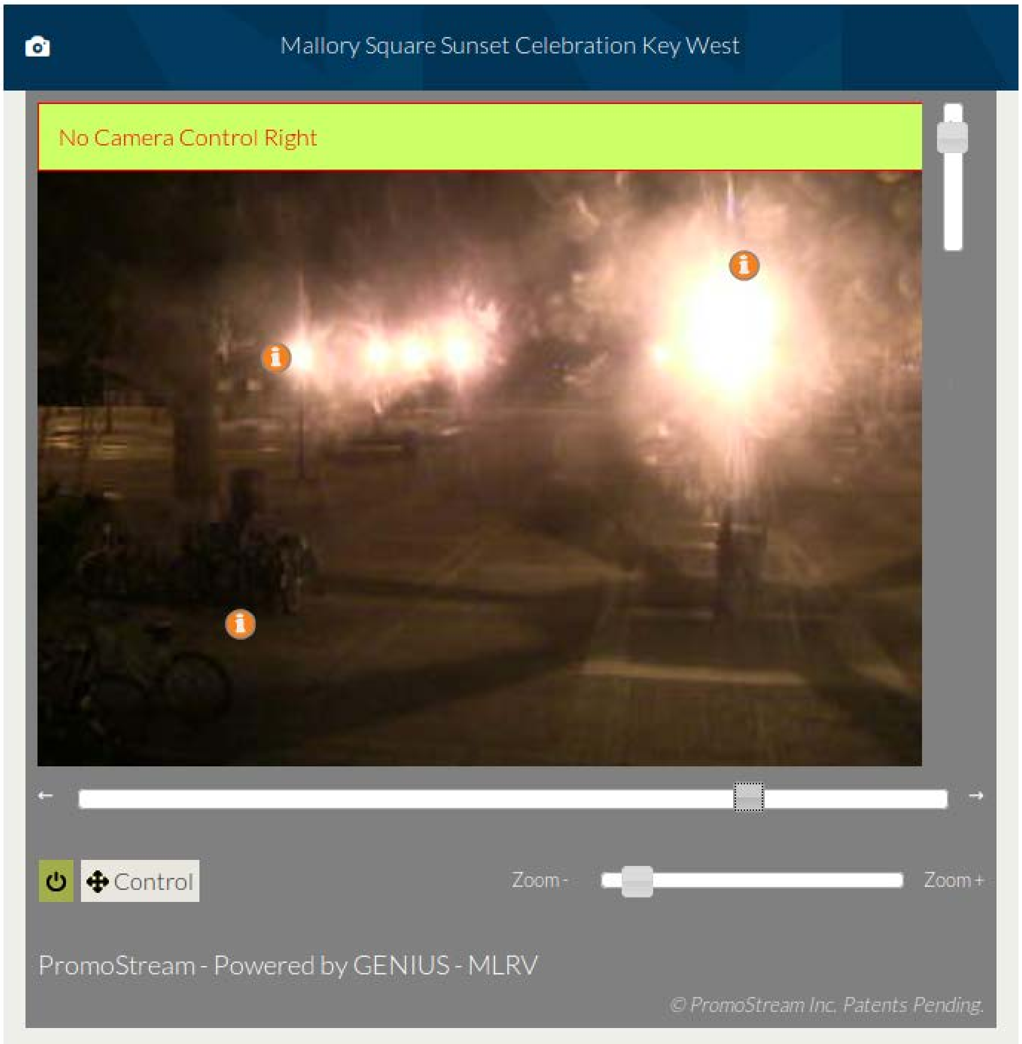


Figure 7: Mallory Square Key West Web Cam

The Mallory Square web cam[40], shown in Figure 7, is also similar to the current project. The camera's horizontal and vertical orientation, along with zoom can be controlled by a user. However, the Mallory web cam does not have an obvious user management system. At times during use the camera appears to be under more than one person's control at a time, though there are also times where access is not allowed with a "No Camera Control Right" message appearing when attempts are made to control where the camera is directed. This web cam also differs from the current project in that there are no saved locations to access.

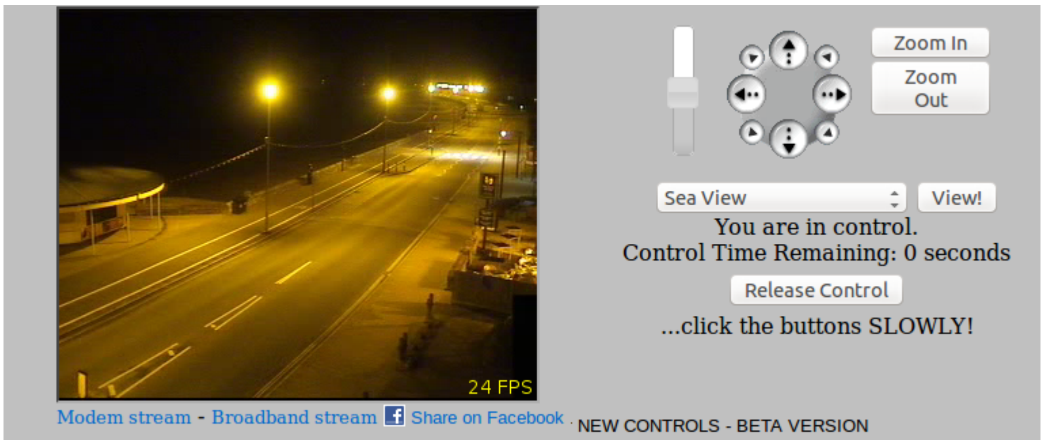


Figure 8: The Exmouth Seafront Beach Webcam.

The Exmouth web cam[17], shown in Figure 8, has some similarities to the current project. It has horizontal and vertical camera orientation and zoom controls like the current project. It also has saved locations accessible via a drop down box. It has several differences from the current project. One difference is that user management is almost non-existent in the Exmouth interface. While it does have a simple queue for requesting access when more than one user is viewing the feed, it does not have any of the current project's advanced scheduling or user creation capabilities. There are also no user specific location saving capabilities like in the current project.

4 Objective/Specification development

4.1 User Requirements

The main customers of the product are Tom Moylan and other scientists at the pier using our equipment for research purposes. Another customer base for this product would be elementary school teachers using the system for education purposes. The list of user requirements was developed for the customers and weighed by importance. This list was then used in the Quality Function Deployment table to compare with the engineering specifications. The user requirements are listed and explained below:

- **Does not leak** - this refers to the pod being able to remain dry on the inside for a reasonable amount of time while underwater.
- **Easy to use** - this requirement affects both sets of customers. The school teacher would want the user interface to be simple and easily control the pod and camera. The researchers at the pier would like the physical system to be easy to work with as well.
- **Resistant to salt exposure** - this requirement is weighed heavily for Tom as the system will be spending a significant amount of time underwater.
- **Prolonged UV exposure** - the product will be exposed to solar radiation at the pier and the components need to remain unaffected after prolonged exposure.
- **Easy to clean** - this requirement refers to the ease with which the maintenance crew can keep the pod and track system clean.
- **Remotely accessible** - this requirement is the ability to be able to control the pod and camera position from a remote computer and also stream live video.
- **Precise and accurate positioning** - this requirement is the ability to control the vertical position of the pod along the length of the track with accuracy.

- **Modularity** - the modularity is the the ability to change or replace subsystems in the pod relatively quickly, and without affecting other subsystems' functions in any way.
- **Quality underwater visibility** - this requirement is based on the user's wish to view the video with some clarity and be able to identify different forms of sea life.
- **Safe to use** - the pod system has to be reliable and secure. The system should not fail when under operation by a user and also not cause any harm.
- **Smooth motion** - the image that is being captured by the camera should not be shaky, which is ensured by having the system move in a smooth fashion so as to avoid jerks.
- **Account management** - this is the ability for users to create accounts where they can store recorded images and videos and view them at later times.
- **General Use by public** - there should be times when the system is available for use by anyone over the Internet when not being actively used for research or education.
- **Create and Use Subroutines** - researchers and educators should be able to schedule camera motion routines to repeat observations for studies into particular wildlife or sea depths.

4.2 Engineering requirements

The user requirements were used to formulate engineering specifications for the project. The specifications were compared with the user requirements using a QFD to ensure agreement between the two. Table 1 lists the specifications that the project is going to meet. The table also lists the likelihood of each specification being met; the risk. The risks are labeled L, M or H for low risk, medium risk, and high risk respectively.

Table 1: Engineering specifications derived from the user requirements listed in the QFD.

Spec. #	Description	Target with Units	Tolerance	Risk	Compliance
1	Submersibility	11 Meters	Required	L	T
2	Data Transfer Rate	1Mb/s	Max.	M	T
3	Time to Change Component	2hours	Max.	M	T
4	Data Storage Capacity	500 GB	Max.	L	I
5	Camera Span Range	60°	Max.	L	I, T
6	Video Quality	1080 p	Required	L	I
7	Video Feedback Delay	3 seconds	Max.	M	T
8	Vertical Speed	0.5m/s	Max.	H	A
9	Concurrent Controlling users	1 user	Required	L	I
10	System Compatibility	Mac, Windows, etc	Min.	M	T
11	Lighting Intensity	2000 Lumens	Min.	L	I
12	System Complexity	# of components	Min.	L	I
13	Tension Drop Standby	30 N	Min.	H	A
14	Automated Freshwater Rinse	Twice a Day	Required	L	T
15	Reliability	99.7% up time	Min.	M	T
16	Overhead	30%	Max.	M	A,T

These engineering specifications are defined as follows:

- **Submersibility** - Refers to the pod's ability to remain submerged at the designated depth without failure/leaking. The pod will be tested at much greater depths for duration ranging from 10 minutes to an hour.
- **Data Transfer Rate** - This requirement has been set due to the pier only having access to a T1 line. This requirement may change as the Pier upgrades it's Internet infrastructure.
- **Data Storage Capacity** - This specification is tentative. It will

change as more progress is made. It is dependent on whether the clients intend to store large amounts video during use, and how successful the device becomes with the public.

- **Time to Change Component** - This specification is meant to ensure major components can be quickly replaced, upgraded or cleaned so there is little downtime and the system can be easily improved and expanded upon. User tests with personnel at the pier will ensure this specification is met.
- **Camera Span Range** - This specification determines the field of view of the camera and it's ability to pan from side to side. It will be tested through observation.
- **Video Quality** - The video quality is set to be high definition for the users to be able to see clear images. This will be tested via observation and user studies.
- **Vertical Speed** - This specification is how quickly the camera moves up on down in a controlled manner up and down the beam. This will be measured with a test using some kind of speed sensor.
- **Concurrent Controlling User** - This specification is set to prevent multiple users from controlling the device simultaneously and thus ruining the user's experience. This will be tested by having multiple users attempt to interact with the device, and verifying that only one is able to attain the authority to control it.
- **System Compatibility** - The system will be used by users across multiple platforms. It is important that they are able to access the pod regardless of their choice of device.
- **Lighting Intensity** - This specification will be investigated through trial and error and research to find an ideal brightness for well-exposed video and still images. This will be verified with a luminosity meter.
- **System complexity** - This specification is to keep the number of parts in the system, both mechanical and software, to a minimum in order to keep it easily manageable and more reliable.

- **Tension Drop Standby** - This specification is to ensure the pod goes into standby mode when tension drops to an abnormally low level which may indicate inclement weather. This will ensure the pod endures as little damage as possible in such situations
- **Automated Freshwater Rinse** - This specification is important in maintaining the longevity of the device. The device will be removed from the ocean at least twice a day and rinsed with freshwater to prevent organisms from growing on the pod. This will be a routine and will be tested by running the routine and observing that the pod is removed and rinsed.
- **Reliability** - This specification is to ensure the maximum amount of up-time and that maintenance does not interfere substantially with the primary project goals.
- **Overhead** - This is the excess computation time and other resources required to communicate between the end user and the pod motor controller.

5 Design Development

5.1 Conceptual Designs

Four main subsystems were focused on during the idea generation phase for the conceptual designs; User Interface, ServoPak to Controller Interface, Pod Electronics Control, and Web Back-end. A flowchart of how the subsystems interconnect is shown in Figure 9.

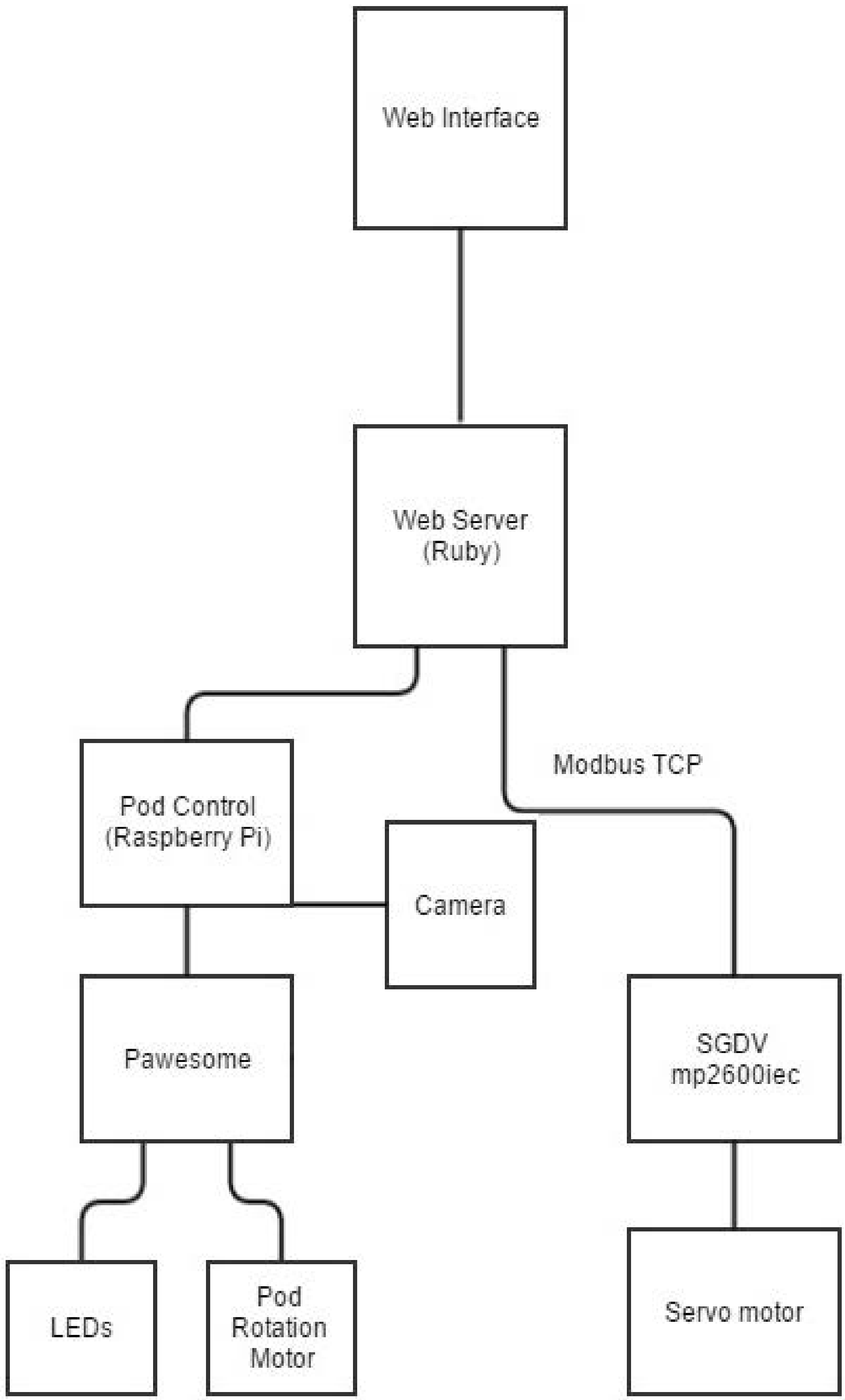


Figure 9: Simple block diagram of subsystem connections with top concepts selected for each subsystem.

The description of the concepts and the selection of the top concept foe each of the subsystems is discussed in the following sections.

5.1.1 User Interface

The user interface will allow the end user to view the video feed from the camera, control the pod location and camera angle, save videos, and other features depending on the user permissions. The options for the end user interface are listed below.

1. **Web interface**
   A web-based interface would allow the user to access the website and control the camera using any device with an Internet connection and a compatible web browser.

2. **Mobile Application**
   A mobile application affords the user flexibility and portability. Users can access their accounts and stream video from anywhere using their mobile devices or tablets.

3. **Desktop Application**
   A desktop application can be installed on a computer of the user's choice. It has the benefit of being exposed to less security risks and it can be made more efficient than running on a browser.

5.1.2 ServoPak to controller Interface

Yaskawa has donated a ServoPack and Servo to the Pier Project. The ServoPack is a SGDV MP2600iec which acts as both an amplifier and a PLC (Programmable Logic controller). The 2600iec is programmable using iec 61131-3 standard languages including function block diagram[10]. The 2600iec comes with out of the box implementations of PLCopen standard function blocks[8]. function blocks consist of inputs and outputs that are linked to addresses in the PLC's memory. These memory addresses can

be accessed via various communication protocols that are supported by the 2600iec. Three of these protocols are implemented over a cat5 line and one is based on analog and digital i/o.

1. **OPC**

   OPC was built using a large number of Microsoft proprietary technologies including: Activex, COM/DCOM (Distributed Component Object Model), XML, and .NET. OPC originally stood for OLE for Process Control. OLE being another Microsoft proprietary technology that stood for Object Linking and embedding. This protocol required all devices participating to be implementing an ioleobject interface, which is an object model that allows the device to communicate it's functionality to requesting clients. OPC has been continuously updated and changed since its conception. It was criticized often for being unstable but has since been touted as more secure as it has evolved. OPC, because of its object abstraction of devices, is suited for large scale data acquisition systems in which many different OPC Slaves/Servers are connected and taking measurements.

2. **Ethernet IP**
   Ethernet Industrial Protocol is an application layer protocol that is transported inside TCP/UDP packets. It requires that all devices attached to the network display their information as objects to be accessed for input/output requests. There are objects that are required to be implemented in all devices, these objects are usually for attaining information from the device, like serial numbers, vendor IDs, and MAC addresses. Then there are application layer objects that are implemented on devices based upon there functionality. These Objects are structured based on CIP or the Common Industrial protocol. Ethernet is a complex and very robust communication protocol[36].

3. **Modbus TCP**
   Modbus TCP is an Application layer protocol that is transported inside TCP packets. It allows for communicating various read and write commands to individual, or multiple addressed registers, or coils. Registers are 16 bit values in the servers memory and coils are 1-bit. There

are input registers/coils which can be both read from and written to, and there are holding register and status coils which can only be read from. These register and coil addresses, can be linked to input/output signals of function block programs implemented on a PLC. "The MODBUS TCP/IP protocol is being published as a ('de-facto') automation standard"[36].

4. **Analog/ Pulse Train**
   The SGDV mp2600iec has 15 programmable digital inputs, 11 programmable digital outputs[10], and 16 bit 10 volt Analog input and output. These digital input and outputs are monitored and written to by the mp2600iec and can be written and read from by a server with digital i/o capabilities.

5.1.3 Pod Electronics Control

The Pod electronics control refers to the components housed inside the Pod. The lighting intensity of lights and the movements of the camera need to be controlled based on user inputs received. The video from the camera should also be compressed and streamed back to the user.

1. **Pawesome Board**
   The Pawesome Board was custom designed by a previous Pier Profiling team. While it is not as standards compliant or flexible as some of the other options we explored, it's strong selling point is the purpose built design specific to the current project.

2. **TI Micro controller**
   Texas Instruments has a line of low power, inexpensive microcontrollers that could take the place of the Pawesome board in the current project.

3. **RaspberryPi**
   A RaspberryPi is a small, single processor computer, and has a wide range of tools available for python, which is one of the options for the web-back end programming language. It also has a video controller that is capable of high definition resolutions.

4. **Arduino Micro controller**
   Arduino is a family of single board microcontrollers with open source hardware. They are fairly inexpensive and common, and have multiple libraries that can be downloaded for free.

5.1.4 Web Back-end

1. **PHP**
   "PHP is a popular general-purpose scripting language that is especially suited to web development"[29].

2. **Python**
   "Python is a programming language that lets you work quickly and integrate systems more effectively"[34].

3. **Ruby**
   "Ruby is a dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write"[38].

5.2 Top Concept Selection

The ideas for the four subsystems were weighed against the engineering requirements and a top concept was selected for each subsystem. The engineering specifications were given weights on their importance. Only the engineering specifications that are directly affected by the four subsystems were used in the decision matrix. The other engineering specifications for the project were not given any weight.

Each subsystem had a datum against which all the ideas were compared. Due the lack of a product that does something similar to our project, one of the ideas in each subsystem was chosen as the datum for that subsystem. Generally, the most common or widely used option was used for the datum. Each idea that was similar to the datum for a certain specification, it received

a score of ‘S’. If the idea was better than the datum, it received a score of ‘+’, and if it was worse then it received a score of ‘-’.

5.2.1 User Interface

The choices for the user interface included a web interface, a mobile app, or a desktop app. When comparing these options, the web interface was chosen as the datum against which everything will be compared. A web interface is very common and many other remotely operated camera systems use a web interface. The reduced decision matrix for the user interface is shown in Table 2.

Table 2: Decision matrix reduced to show the ideas and top concept for the user interface.

System Functions	Potential Solutions	Data Transfer Rate	Time to change component	Video quality	Video feedback delay	System Compatibility	System Complexity	Reliability	Sum +	Sum -	Sum S	Total
Specification Weight		10	9	9	10	7	15	19				
User Interface	Web Interface	S	S	S	S	S	S	S	0	0	79	24
	Mobile App	-	-	-	-	-	-	+	19	60	0	-41
	Desktop App	S	-	S	+	-	-	+	29	31	19	3.7

Since the web interface was the datum, it received a score of ‘S’ throughout. Both the mobile app and the desktop app were given a score of ‘+’ for reliability since they would reside on the user’s device instead of being hosted on a web server.

The mobile app rated worse than the web interface for the rest of the engineering requirements. It would be difficult to design a mobile app that is compatible across all device types, which adds to the complexity of the system. Also, changing the interface would require the users to update their apps before any changes can take effect. Thus it received a score of '-' for system compatibility, complexity and time to change component. The data transfer rate and video feedback delay depend on the mobile phone or tablet's connectivity, which in turn would also affect the quality of the video being received by the user. Since it cannot be guaranteed that the user will always be able to connect to high speed Internet, the mobile app received a score of '-' for data transfer rate, video quality, and video feedback delay.

The desktop app was rated worse than the web interface for time to change component, system compatibility, and system complexity for the same reasons as the mobile app. Designing a compatible app would add to the complexity and also require longer for changes to take effect. The desktop app was given a score of 'S' for data transfer rate and video quality since the app would most likely be using the same Internet connection as the browser used for the web interface. It received a score of '+' for video feedback delay because the processor of the computer can be used by the app to decrease processing time and result in a faster response time[48].

After adding the total weighted scores of each potential solution, it was determined that the web interface was the best choice for the user interface. It requires fewer components and it can be compatible across various platforms and devices with relative ease.

5.2.2 ServoPak to Controller Interface

Table 3: Decision matrix reduced to show the ideas and top concept for the ServoPak to controller interface.

System Functions	Potential Solutions	Time to change component	System Compatibility	System Complexity	Overhead	Reliability	Sum +	Sum -	Sum S	Total
Motor to controller interface	OPC	S	S	S	S	S	0	0	55	17
	Ethernet IP	+	+	+	-	+	50	5	0	45
	Modbus TCP	+	+	+	S	+	50	0	5	52
	Analog/Pulse Train	-	S	-	+	-	5	43	7	-36

The ServoPak to Controller interface needs to enable our server to communicate instructions from the web client to the ServoPak to change the pod's depth. The Server will be running in the language selected via the decision matrix for the web back-end sub-system, shown in Table 5. The ServoPak supports, Modbus TCP, Ethernet Industrial Protocol, OPC, and Analog and digital I/O.

OPC was dismissed because choosing it would practically force us to use a windows based server. It is also somewhat more complicated and has been known to have trouble with hardware and embedded systems. Ethernet IP is very robust, but it is also more complicated and requires more overhead[35]. It would also take much longer to implement. Ethernet IP would have been a good option had the network contained more devices in a more industrial setting.

Modbus TCP was chosen because it appears to be the least complex and

most open of all the options. it would just require an Ethernet connection to the server and it has a wide tolerance for network capabilities[36]. There are open source implementations for Modbus TCP in Python, Ruby, and PHP[35]. Modbus TCP can be ported between any server operating system or architecture. It requires very little overhead because it's functionality is to read and write bits to and from addressed locations.

This subsystem will thus consist of a mp2600iec programmed using PLCOpen function blocks like the one seen in Figure 10.

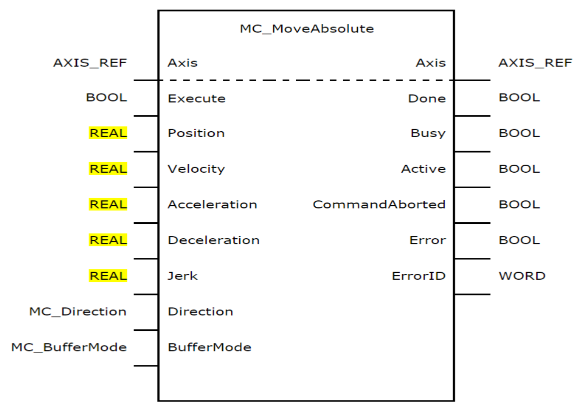


Figure 10: Example PLCOpen Function Block MC MoveAbsolute[31].

The inputs and outputs of the function blocks can either be linked to other function block inputs/outputs or they can be linked to addresses in memory.

Name	Type	Usage	Description	Address
⊟ <SGDV Rotary> - Sigma-V Rotary Servo Amplifier - 1:1 (* Modify Variable Names, Not Group Name. *)				
AX1_SI1_POT	BOOL	VAR_GLOBAL	POT, default on pin #7, configurable by Pn50A.3	%IX53248.0
AX1_SI2_NOT	BOOL	VAR_GLOBAL	NOT, default on pin #8, configurable by Pn50B.0	%IX53248.1
AX1_SI3_DEC	BOOL	VAR_GLOBAL	DEC, default on pin #9, configurable by Pn511.0	%IX53248.2
AX1_SI4_EXT1	BOOL	VAR_GLOBAL	EXT1, default on pin #10, configurable by Pn511.1	%IX53248.6
AX1_SI5_EXT2	BOOL	VAR_GLOBAL	EXT2, default on pin #11, configurable by Pn511.2	%IX53248.7
AX1_SI6_EXT3	BOOL	VAR_GLOBAL	EXT3, default on pin #12, configurable by Pn511.3	%IX53249.0
AX1_BRK	BOOL	VAR_GLOBAL	Brake Output Status	%IX53249.1
AX1_HBB	BOOL	VAR_GLOBAL	HBB, Stop Signal Input	%IX53249.2
AX1_SI0_IO12	BOOL	VAR_GLOBAL	Configurable by Pn81E.0, default is unallocated	%IX53249.4
AX1_SI1_IO13	BOOL	VAR_GLOBAL	Configurable by Pn81E.1, default is unallocated	%IX53249.5
AX1_SI2_IO14	BOOL	VAR_GLOBAL	Configurable by Pn81E.2, default is unallocated	%IX53249.6
AX1_SI3_IO15	BOOL	VAR_GLOBAL	Configurable by Pn81E.3, default is unallocated	%IX53249.7

Figure 11: Variables are mapped to an Address[28].

The MP2600iec will map Variable addresses to Modbus addresses; this will enable a Modbus client to access these addresses with read and write commands.

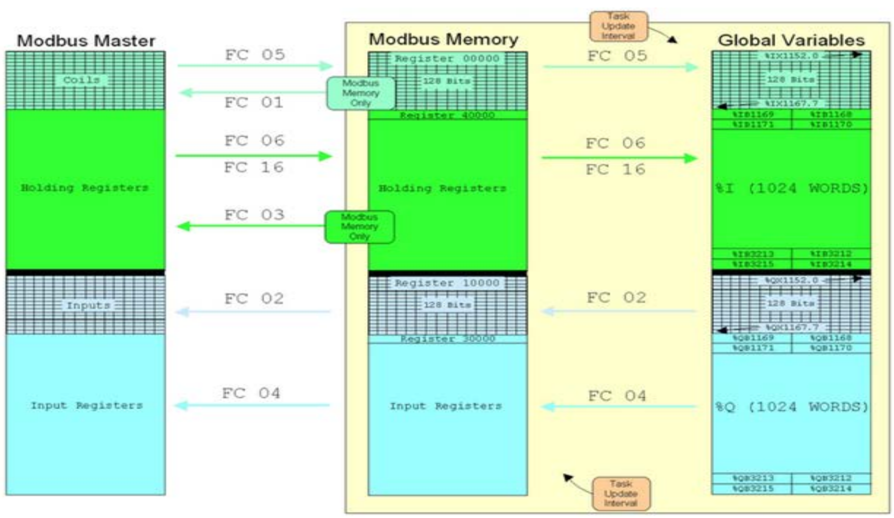


Figure 12: Modbus Registers and coils are linked to variable addresses[28].

The Pier Server will act as a Modbus client and will be able to interface with the ServoPack via the following Modbus functions supported by the mp2600iec.

Function Code	Description	Modbus Address Range (On Slave)
1	Read Coils	00001 to 10000
2	Read Inputs	10001 to 20000
3	Read Holding Registers	40001 to 50000
4	Read Input Registers	30001 to 40000
5	Write Single Coil	00001 to 10000
6	Write Single Register	40001 to 50000
16	Write Multiple Registers	40001 to 50000

Figure 13: The Modbus Functions that are supported by the SGDV mp2600iec[28].

5.2.3 Pod Electronics Control

The potential solutions for the pod electronics control included the Pawesome board, a TI micro-controller, a RaspberryPi, and an Arduino micro-controller. The Pawesome board was chosen as the datum for this subsystem. It was designed by the previous team and installed in the pod to control the electronics. The comparison was made with other microcontrollers to determine if it was the best choice. Table 4 shows the reduced decision matrix for the pod electronics control subsystem.

Using a Texas Instruments microcontroller or an Arduino microcontroller in place of the Pawesome board was determined to not be very beneficial. First of all the microcontrollers considered in this comparison have less data throughput than the Pawesome board, which would inhibit the data transfer rate and possibly lead to larger amounts of lag time between issuing a command and command execution. Not to mention, removing the Pawesome board and adjusting all the pod connections would bring system reliability

into question compared to the Pawesome board which is already proven to work. Adjusting the wiring of the pod to interface with a microcontroller and physically connect to it would also increase system complexity slightly, which is to be avoided at all costs.

The replacement of the Pawesome board with a RaspberryPi computer was also considered and ultimately discarded. Using the RaspberryPi would have mostly the same drawbacks as either the TI microcontroller or the Arduino microcontroller. The only possible benefit to using a RaspberryPi computer would be a higher data throughput, though with the already proven system including the Pawesome, this would not justify such a drastic change.

Table 4: Decision matrix reduced to show the ideas and top concept for the Pod electronics control.

System Functions	Potential Solutions	Data Transfer Rate	Video feedback delay	System Complexity	Reliability	Sum +	Sum -	Sum S	Total
Pod Electronics Control	Pawesome Board	S	S	S	S	0	0	54	16
	TI Microcontroller	-	S	-	-	0	44	10	-41
	Raspberry Pi	+	S	-	-	10	34	10	-21
	Arduino	-	S	-	-	0	44	10	-41

5.2.4 Web Back-end

Table 5: Decision matrix reduced to show the ideas and top concept for the web back-end.

System Functions	Potential Solutions	Data Transfer Rate	Time to change component	Data storage capacity	Video quality	Video feedback delay	Concurrent controlling users	System Complexity	Reliability	Sum +	Sum -	Sum S	Total
Web back-end	PHP	S	S	S	S	S	S	S	S	0	0	88	26
	Python	S	+	S	S	+	S	S	S	19	0	69	40
	Ruby	S	+	S	S	+	S	+	+	53	0	35	64

For the web back-end, the following three languages were compared; PHP, Python, and Ruby. For Python, the Django framework was focused on, and for Ruby, the focus was on the Ruby on Rails framework, as they are some of the popular frameworks[47]. The following criteria were used when comparing these languages with one another; the purpose the language is intended for, ease of deployment, video streaming capability, and general popularity in industry. Video streaming and ease of deployment are the most crucial for this project, so the purpose and popularity of the languages were used as a tie breaker. The decision matrix for the subsystem and the top idea are shown in Table 5.

Video streaming is a crucial component of the web application. Therefore, this criteria was weighted the most out of all the others for the web back-end. The easier it is to integrate live streamed video and playback of recorded video, the better the language. PHP video streaming has a variety of options available for video streaming. These options include Youtube Live streaming API and HTML5 video streaming[20][14] to name a few. Django has django-video developed by andrewebdev is one option for Django streaming[19]. However it is recommended to use a dedicated media server when working with Django which include; Flash Media Server, Wowza media server, or

Red5. All of those options, except for Red5, are propriety and require a subscription to use[41]. Ruby on Rails has a couple of options for video streaming, Panda and open-tok. Panda is a video streaming service which requires a subscription to be used. Open-tok is also a streaming service, however there are also server SDK packs available for development on you own server.

Web application deployment is defined as the process of installing the application into server context[13]. The purpose of looking at deployment is to determine how easy it would be for us to transfer our code from a testing server to the production server at the pier. A secondary reason is to determine how easy it would be for another pier to utilize our software, assuming they already have the hardware ready. PHP has various options for server deployment, all of which are highly dependent on the server configuration[46]. One option is to use SVN and Phing, Phing is a PHP build system based on Apache ANT. Another option is to use Github as a method of deployment[42]. This is more involved and is less automated than other options. Django has a deployment checklist for settings that must be set on the Django server beforehand. The checklist can be run using Django's manage.py file for some of the settings, however it does not guarantee it will work for all settings[16]. Ruby on Rails has the advantage of following convention over configuration[37]. This means that there is convention involved when designing server configuration, so it is not completely up to the system administrator as to how the server will be configured. This means that for other piers there is a convention they can rely on for configuring their server to work with our software, rather than having to interpret another team's custom configuration.

PHP is a wildly popular scripting language used primarily on the Internet alongside HTML. It has been developed over the years to handle many use cases with multimedia and web site interactivity. The ground-up development is evident in the varied influences of the language and its syntax[23]. It is also clear there is no strong underlying design principle, and scripts written in the language can easily become unreadable. Python was designed with readability in mind[33]. There is generally one way to write code in Python, which limits the programmers options however allows for another programmer to quickly read and understand a program written in Python. Ruby was designed with flexibility in mind. Unlike Python, there are multiple ways for

a programmer to express the same idea. However, Ruby on Rails follows the convention of least astonishment. This is meant to prevent the user of an interface made in Ruby from becoming confused on ambiguous portions of an interface, which in some languages might require knowledge of the inner workings of the program[32]. Ruby on Rails also follows the DRY paradigm: Don't Repeat Yourself[37], which makes coding for programmers easier.

PHP is the most popular of the three languages, it dominates the Internet with 82% back-end servers running PHP[45]. It first appeared in 1995, with the intention to parse HTML and make C library calls. Python is the least popular web back-end language. It has been in existence since 1991, four years longer than PHP or Ruby. However it is currently used for scientific computing[39], among other non-web related computation. Ruby is the next most popular web back-end language of the three, closely followed by Python. It first appeared in 1995 as well, however it was intended to bring a balance between functional and imperative programming, not necessarily web development.

5.3 Concept Design of User Interface

To select the layout of the web-based user interface, a few key screens were prototyped, shown in Figures 14 - 18 . These prototypes were presented to several testers who were tasked with providing feedback on the appropriateness of the interface for the desired end user types.

5.3.1 Prototype Images

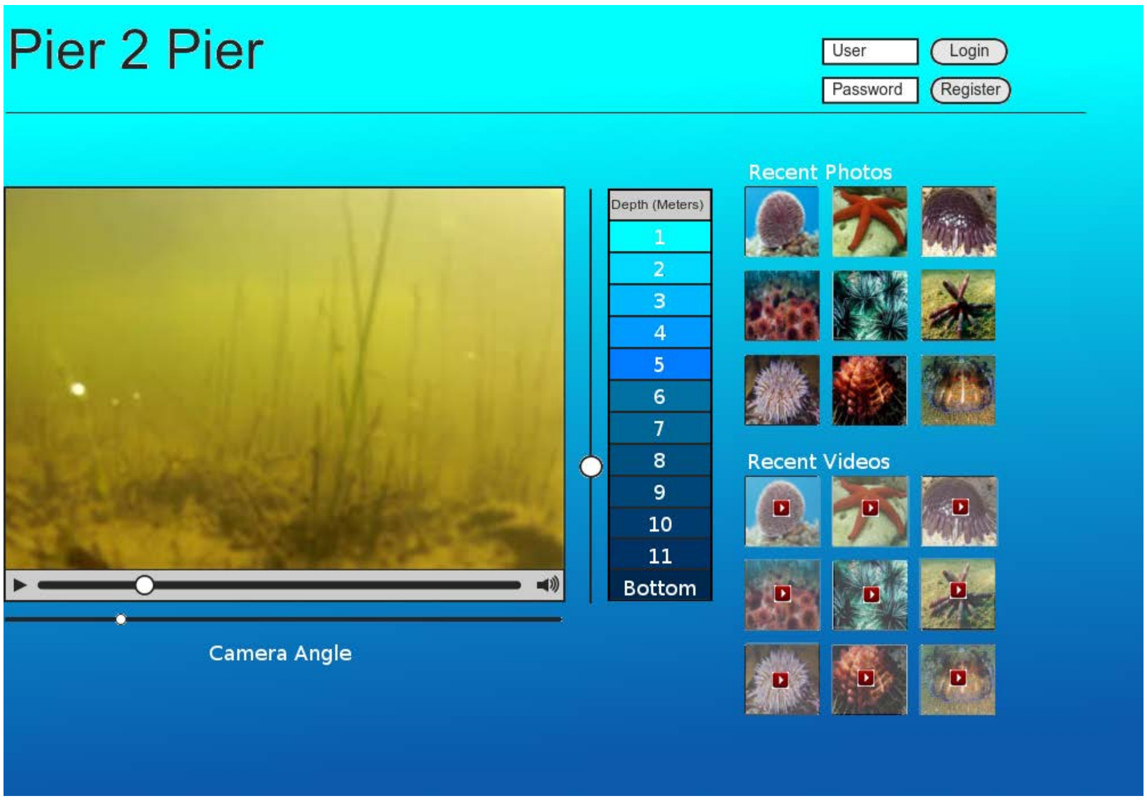


Figure 14: Home screen of web interface.

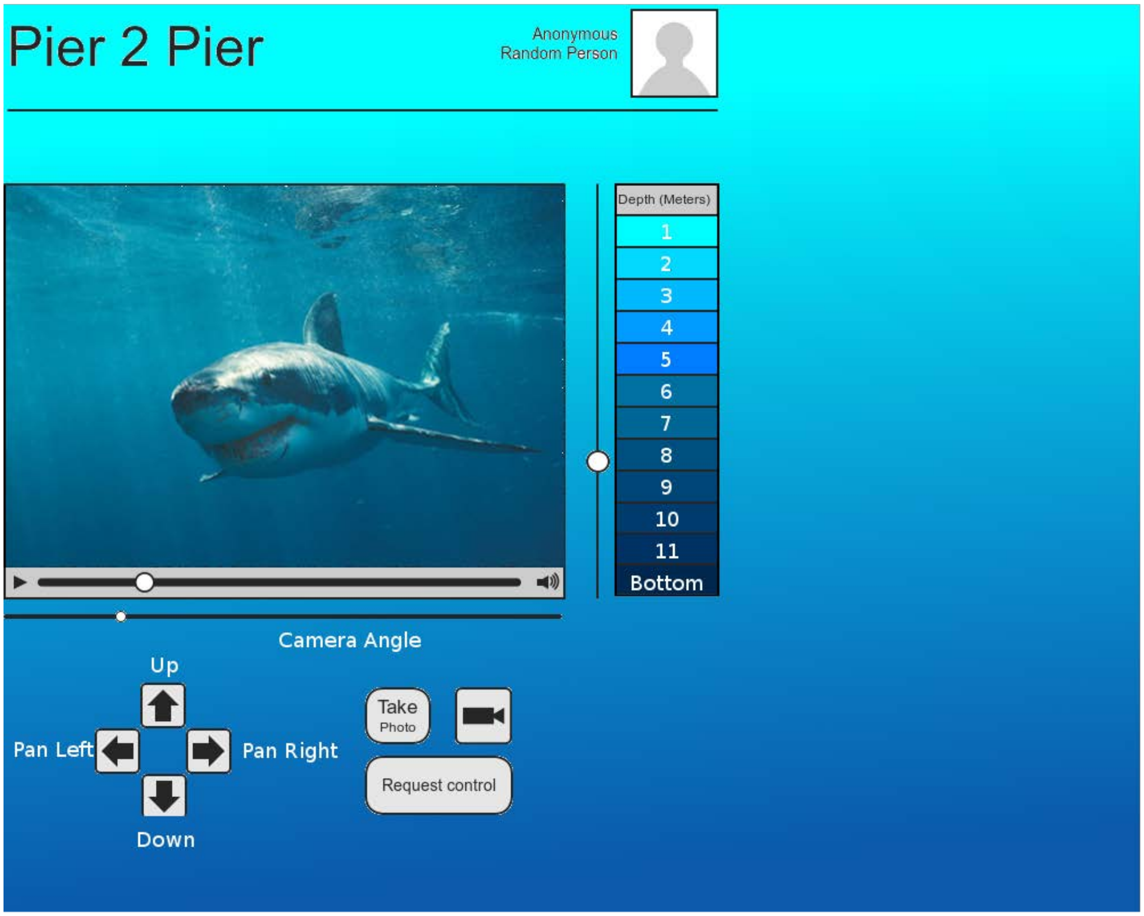


Figure 15: Screen for a user who is logged in.

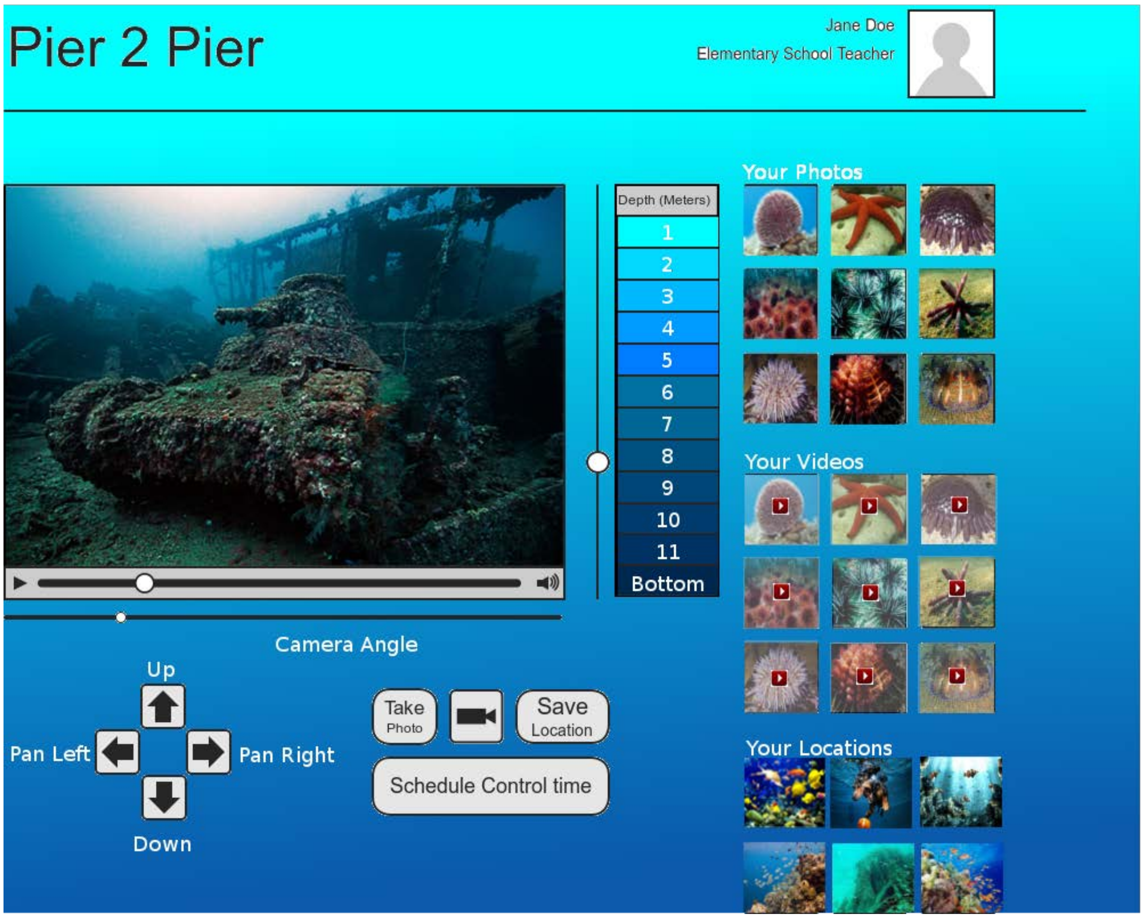


Figure 16: Screen for an elementary school teacher who is logged in.

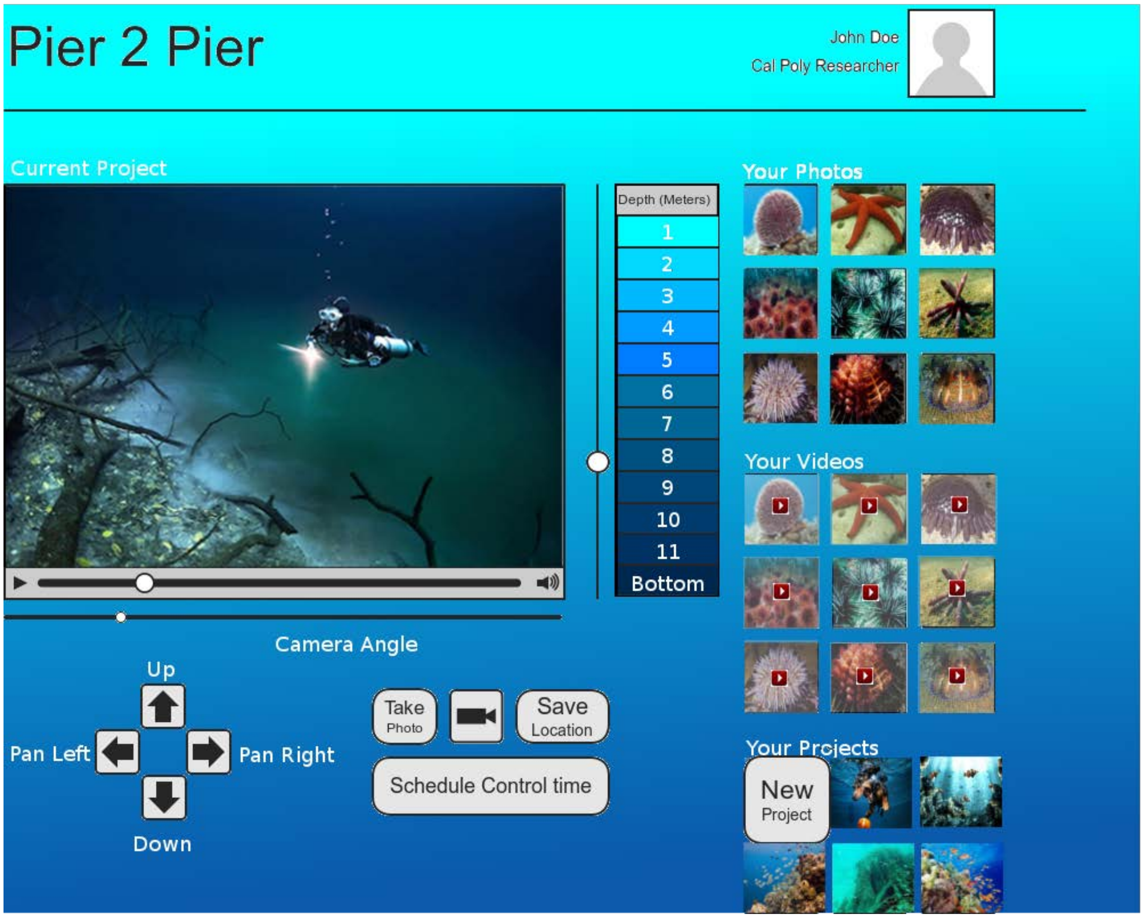


Figure 17: Screen for a researcher who is logged in.

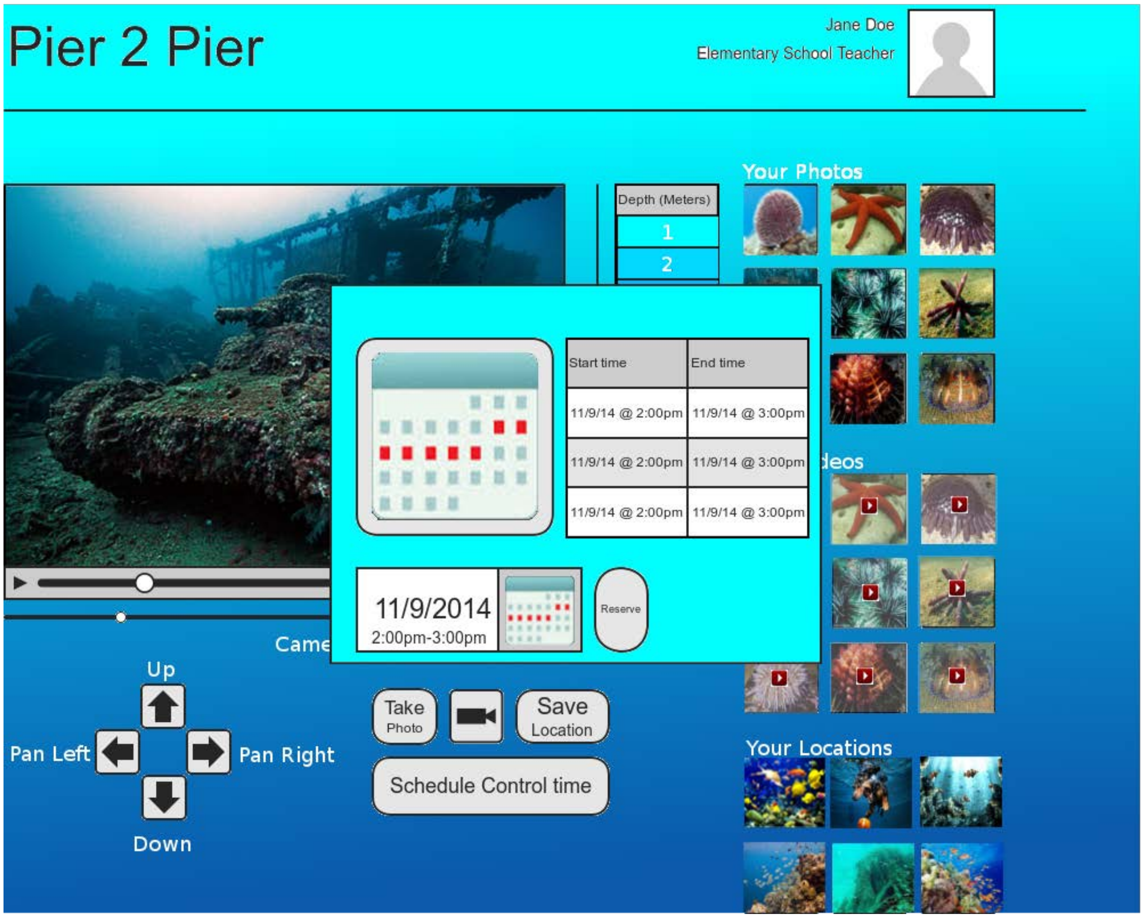


Figure 18: Pop-up screen that appears when scheduling control.

5.3.2 Feedback

The testers were asked to fill a short questionnaire shown in the Figure 19. The results of their feedback were collated and the several repeated concerns and criticisms were noted and applied to the prototype pages. The most common criticisms and complaints are noted in Table 6.

Table 6: Common feedback received for the user interface layout

Feedback	Occurrences
Save Video Button nonintuitive (make more straightforward)	4
Video shouldn't touch left side of screen	3
save location is confusing (file location? pod location? need clarification	3
Too much Blue change color scheme	2
North South East West labels on camera pan (or compass style)	2
Full screen button	2
Sliders are nonintuitive (do they control the pod?)	2
Schedule control should have times available and times reserved	2
Change location of request control button	1
Some means of notifying user that another user is in control of the device	1
Public videos and photos accessible by anybody	1
Change theme/scheme based on profile type	1
Light or zoom control?	1
Map controls to keyboard	1
Buttons should have helpful tool tips	1
Have preset locations	1
Is the video recording or live stream?	1
Add weekly view to scheduler	1

Consider the following user groups who are going to be using the interface and then answer the short feedback questions below:

a. An elementary school science teacher who wants to login to the website and show her students videos and pictures of sea life and have an interactive discussion.

b. A marine researcher who wants to observe different areas of the pier piling and take regularly scheduled photographs and videos of his various areas of interest.

c. A civilian who has created an account and logged in to the website and wishes to control the camera and view the video for recreational use.

d. A first time visitor to the website, viewing the home page.

What are your comments and suggestions about the following aspects of the model:

1. Button placement

2. Graphics Organization (Aesthetics)

3. Functionality(Work Flow / Ease of use)

4. Design Features (e.g. saving location, project folders)

5. Anything else that should be added / deleted / changed

Figure 19: Feedback Questionnaire presented to testers of the interface.

The feedback will be taken into consideration during the next iteration of the interface design. The comments helped in finding flaws in the design and finding a more user friendly design. It was most helpful in identifying parts of the design that were quite confusing. The next iteration of the prototype will be shown to a larger pool of testers, enabling us to collect a bigger range of statistical data to analyze and improve the design.

6 Description of the Final Design

6.1 Overall Description

The project consisted of a means of streaming high definition video over a computer network accessible by most modern web browsers while also providing a means of vertical position control via the web interface. Preliminary methods of saving and going to absolute locations were implemented. A mounting bracket was developed to interface the servo motor with the winch. A web interface was also developed that allows site visitors to control the pod and to view the live feed from the pod.

6.2 Web Interface

6.2.1 Account Creation

Users will be prompted to log in to a previously created account or create an account upon accessing the web site for the pier. When a user elects to create a new account they will be presented with the Account Creation page shown in Figure 20. This page will request basic information about the user, including their email address, desired password, and their purpose in creating the account. Upon completion of this page, the user will be presented with a another page depending on what type of account they are creating. If the user created a general public account, they will be presented with a page telling them a confirmation email has been sent to the email

address they provided to ensure they have access to that email address. If the user chose to create a researcher account or an educator account they will be presented with another page requesting the name of the institution they are representing or working for. The institution name they provide will be checked to see if it is in a list of accepted institutions. If the user is representing an approved institution they will be presented with a page similar to the standard user's email confirmation page, and will be instructed to follow the link sent to them via email to access their account for the first time. However, if the user did not input an approved institution they will be presented with another screen describing the error and suggesting that the user check the institution name they inputted and that they contact their administration or the pier manager for additional assistance.

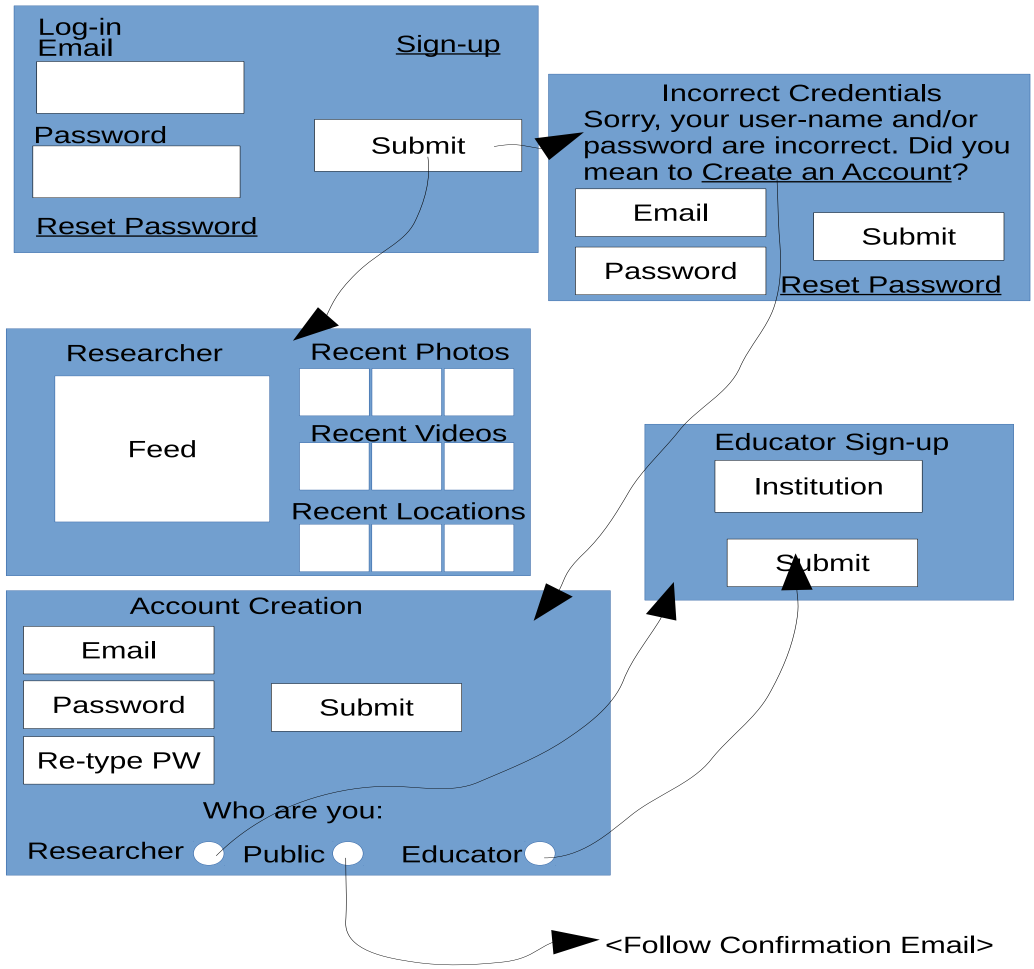


Figure 20: Shows how the user logs in and the steps to creating an account.

6.2.2 Account Log-in

Upon accessing the website for the pier, a registered user will have the ability to sign in on their existing account, as shown on the Log-in page in Figure 20 If they enter credentials corresponding to a previously registered user that has followed the confirmation link sent to their email address, they will be

presented with the initial screen for the user type associated with their user account. An example of the page a researcher would see is provided in the Researcher page shown in Figure 20. On the other hand, if a user inputs credentials for a user that does not exist, they will be presented with the Incorrect Credentials page in Figure 20. From that page, the user can try to re-input their credentials, or follow the link to the Account Creation page and go through the steps outlined in the section "Account Creation" above.

6.2.3 General Public User Type

A user with General Public credentials will have limited access to the functionality of the system. They will be allowed to view a live feed from the pod, view and download recently saved photos, and request brief periods of access to the pod's control system when no one else is using it. A sketch of the basic user interface available to general public users is shown in Figure 21. In this figure the user can see a live video stream from the pod and request pod control for a limited amount of time. In Figure 22 the user has selected a recently saved photo from the Recent Photos pane. This brings the user to a page with a larger view of the photo. In this case, the user had control of the pod, so the pod also travels to the location the selected photo was taken, and adjusts to the angle the photo was taken at. If the user was not in control of the pod and clicked on the photo, they would reach a page only containing the full resolution version of the selected photo.

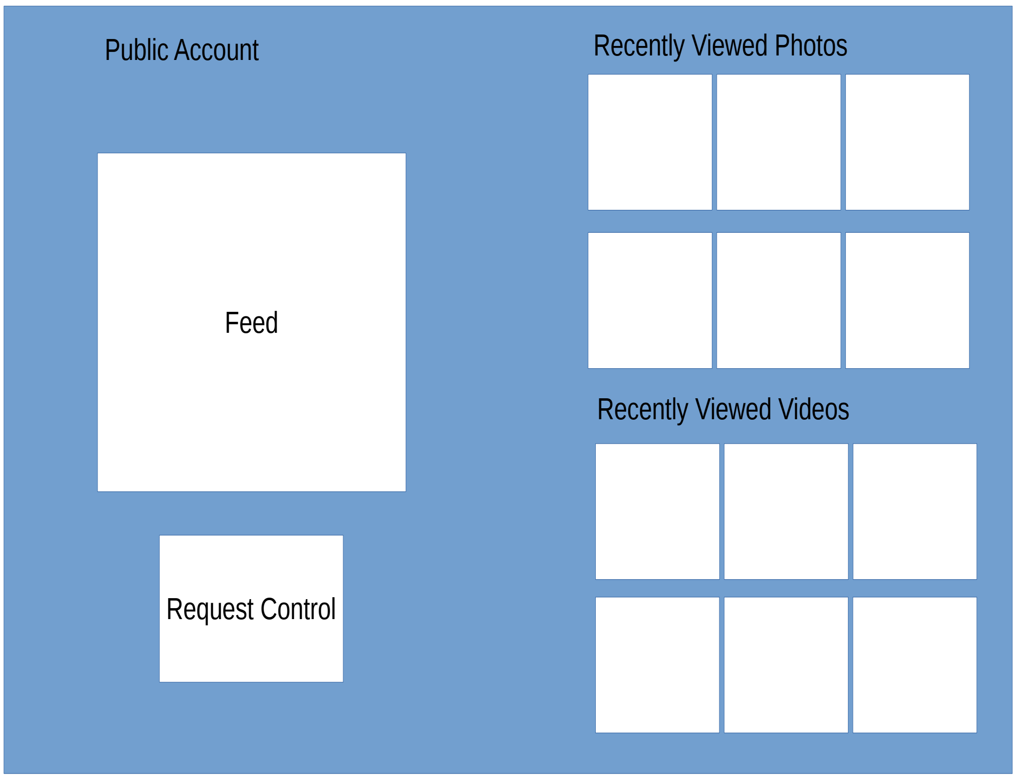


Figure 21: Shows the basic page layout for a general public user.

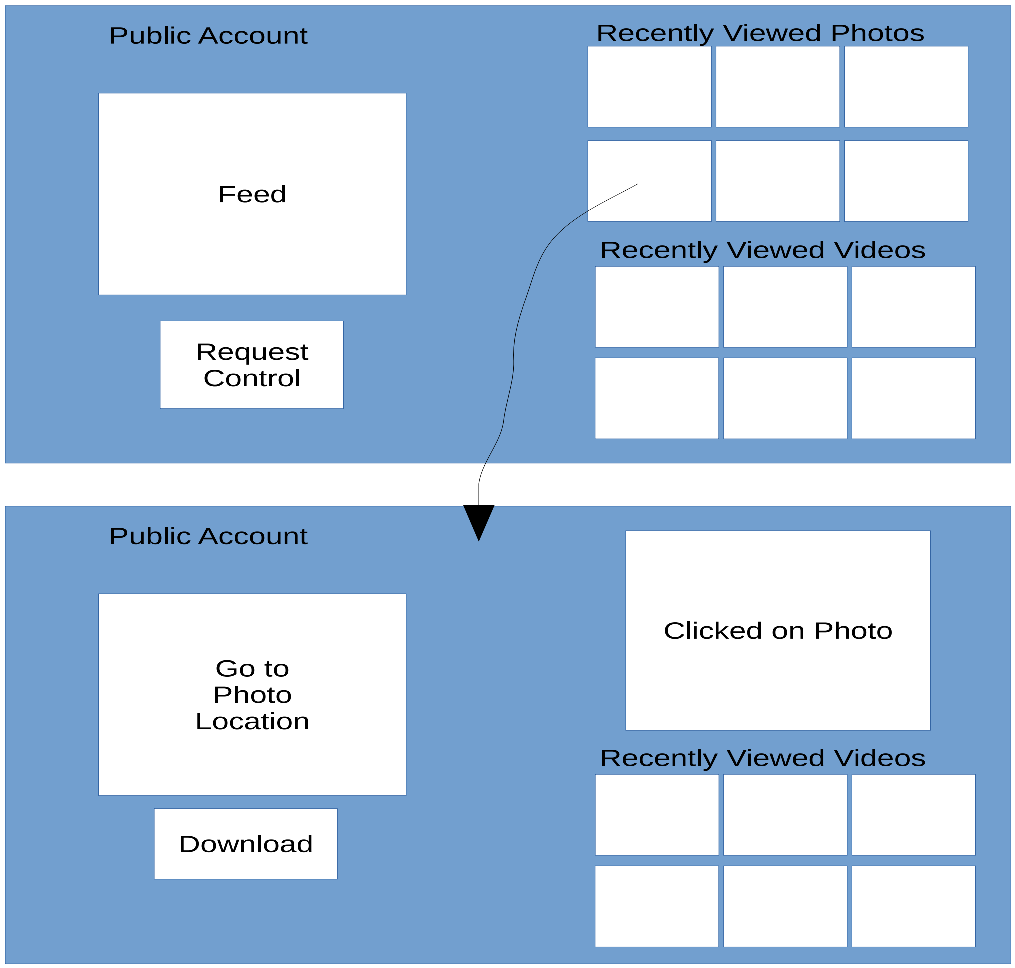


Figure 22: Shows how a general public user can access a recently saved photo.

6.2.4 Educator User Type

The educator user type will have the same capabilities as the general public user type, along with extended pod control time allowances and priority over general public user pod control. In addition, Educators will be able to save and manage pod locations as shown in Figure 23. This figure shows an

educator selecting a saved location. Assuming they have control of the pod, the pod will move to the selected location and adjust to the corresponding angle. The educator can choose to edit the notes associated with the location and see a selection of images that have been taken at that location.

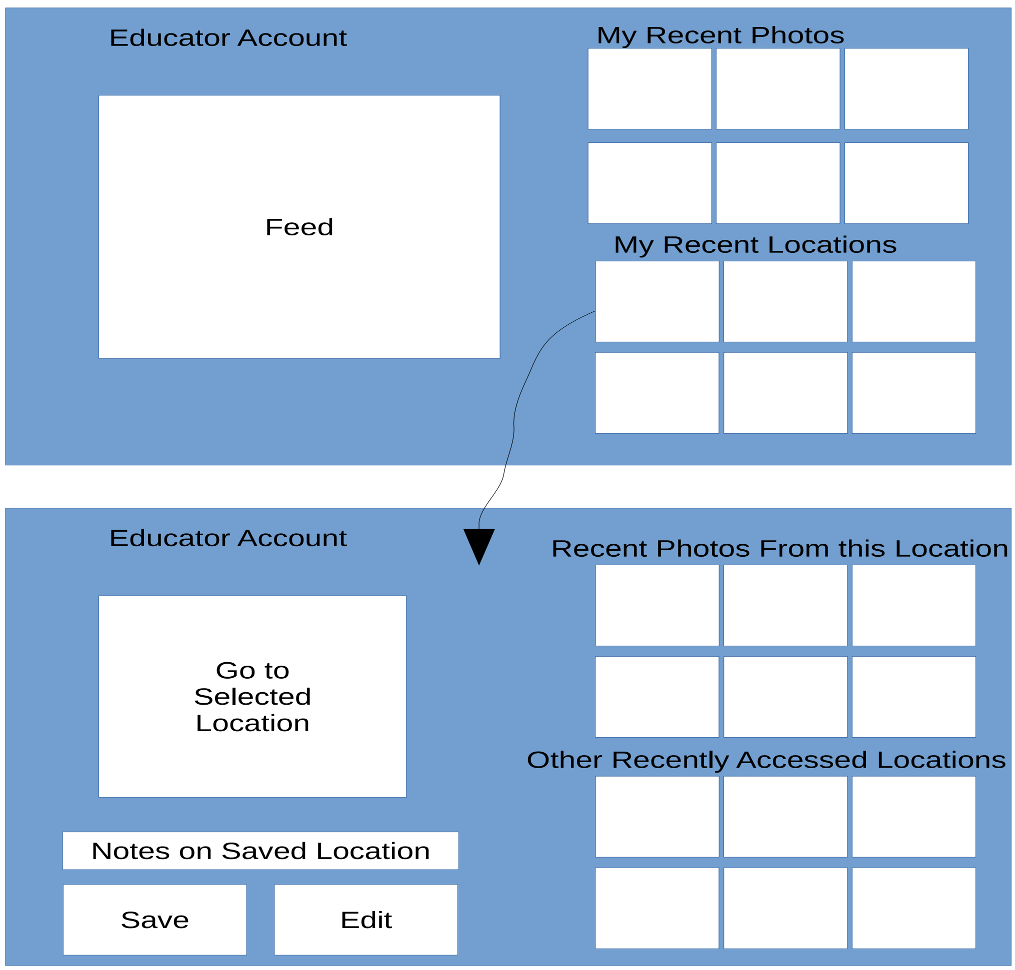


Figure 23: Shows the location management system.

6.2.5 Researcher User Type

Users with the researcher type have the most privileges of all user types. They can use all features of the site available to the general public and educator user types, in addition to extended scheduling capabilities and the ability to record video. Figure 24 shows an example of the researcher type's extended capabilities. Not only can researchers view saved photos from a saved location and edit the location's description, they can also view videos they have recorded at that location previously. The researcher will also have extended capabilities in terms of saving descriptions to and viewing photos, as shown in 25. Once the researcher records a photo, they are presented with the ability to add a description to the photo. As can also be seen in the figure, researchers are able to not only view photos sorted by the time and date they were recorded, but can also sort photos they have recorded based on the depth at which they were taken.

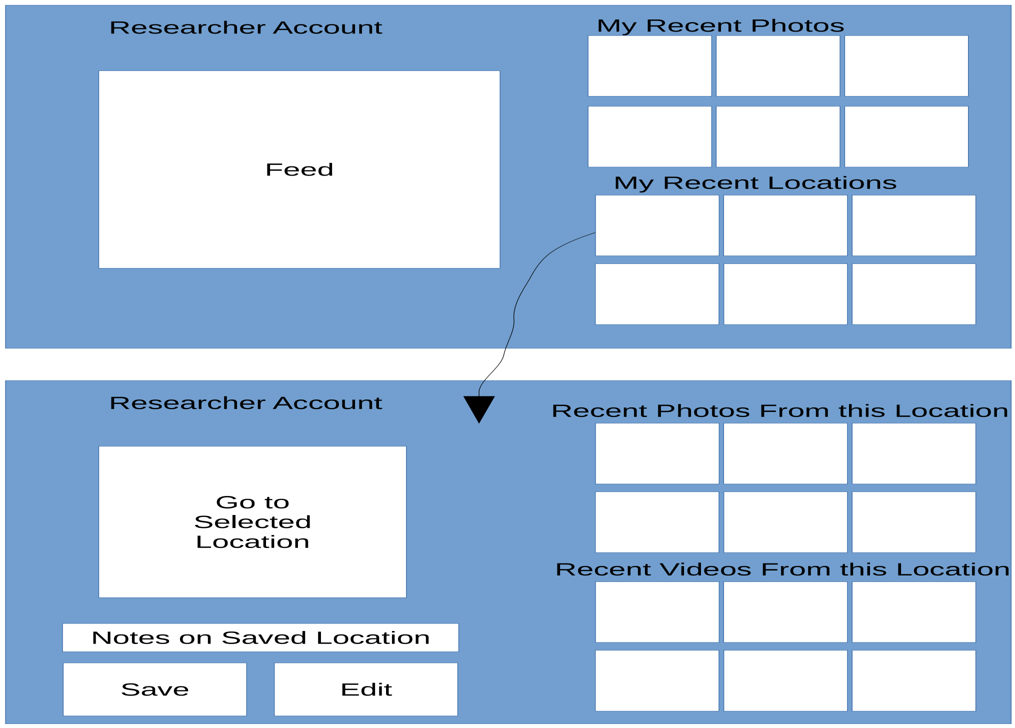


Figure 24: Shows the location management system for researchers.

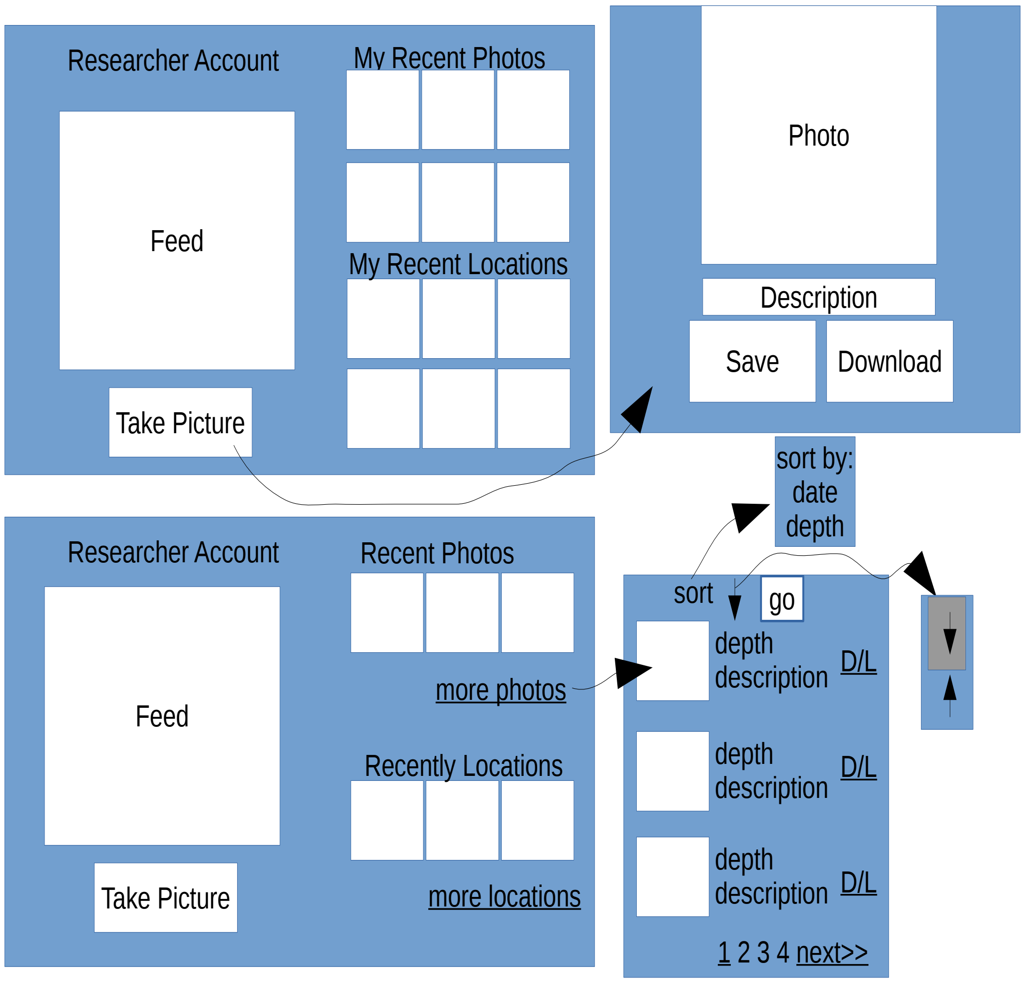


Figure 25: Shows how researchers can sort and add descriptions to photos.

6.2.6 Web site back end

The website was designed using Ruby on Rails as the back-end. The website made use of Controllers and Views for displaying information to the visitor. The Controllers include a welcome, videos, pictures, locations, log-in, and pier. The welcome controller and view is responsible for displaying the

current stream to the website visitor, along with a banner, log-in box, and a list of recent videos and photos. The videos controller is responsible for obtaining and listing the videos on the welcome page, and the pictures and locations controllers behave similarly for the most recently saved pictures and locations. The log-in controller is responsible for authenticating user log-ins, creating new user accounts, and issuing users tokens at login. Finally, the pier controller is designed to send commands to the Cal Poly Pier via HTTP GET and POST requests. This controller is activated by sending an AJAX request from the client to the pier server.

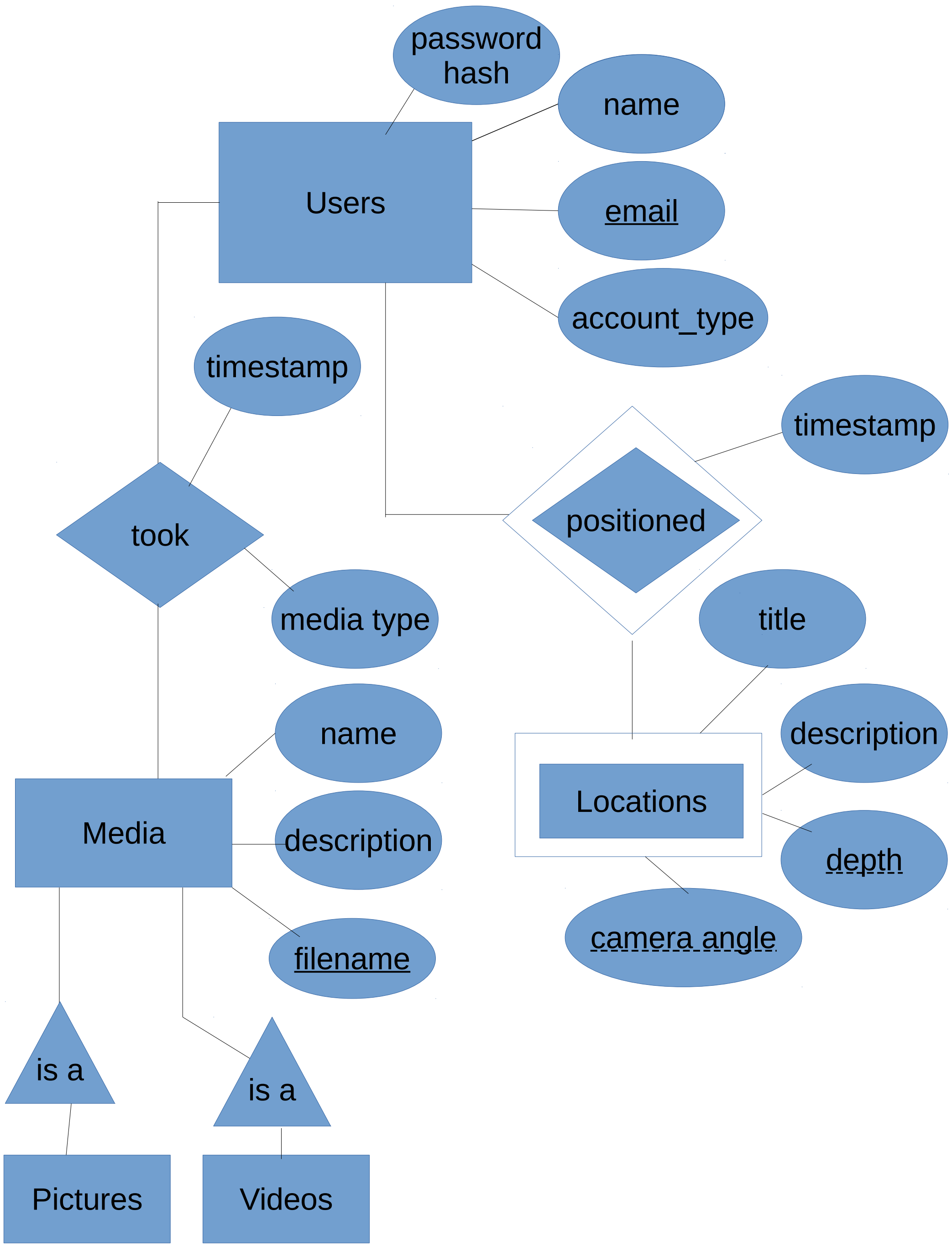


Figure 26: Entity Relationship Diagram for Models

Along with Views and Controllers, there are a couple of Models which our website design makes use of. These include the User, Video, Picture, and Location models. The User model contains information on the users name, email, hashed password, and type of user. The Video and Picture model contains information on the title on the video and a file location on the server. The Location model contains a title for the location, description, and the distance of the location.

The user was given various pod controls, along with standard video recording and picture capturing buttons next to the video stream after logging in. We disabled the buttons for any user who logs in and is not authorized. However, back-end security is also necessary in case a user decides to access the pier controller outside of the web page front-end. For example, without token authentication, a user could visit the URL which the button posts too and achieve the same results as a button press.

To guarantee the requirement that only a single user may control the pod at a given time, the web server kept track of the currently authorized user. This was accomplished by issuing a random token to a user at log-in, keeping track of all the users current tokens on the server side, and storing the issued token as a cookie on the client side. When a user makes an AJAX request to the pier server, the client will post it's token stored as a cookie data to the controller. The pier will then verify that the token received is the authorized token before sending the TCP command to the pier server.

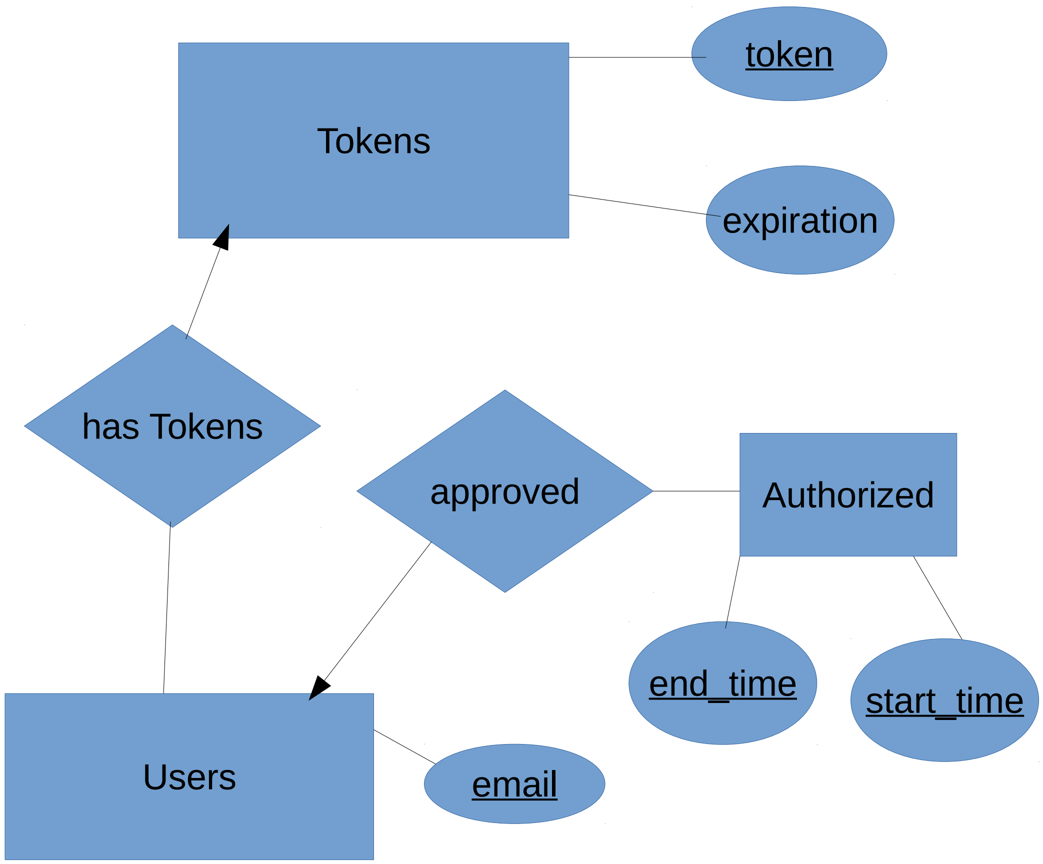


Figure 27: Entity Relationship diagram for pod access control

In order to prevent TCP connections from constantly being opened and closed whenever the user sends a basic command to the pier server, the TCP connection will be opened upon server initialization. This is accomplished by creating a custom ruby gem consisting of a static class which opens the TCP connection, keeps track of the connection, and has several methods for sending messages to the pier server. In the web server initialization file, include the initialization of the ruby gem's static class. This gem will then be accessed by the controller to send specific commands to the pier.

6.3 Video Streaming

Figure 28: The C920 web cam that is mounted in the pod

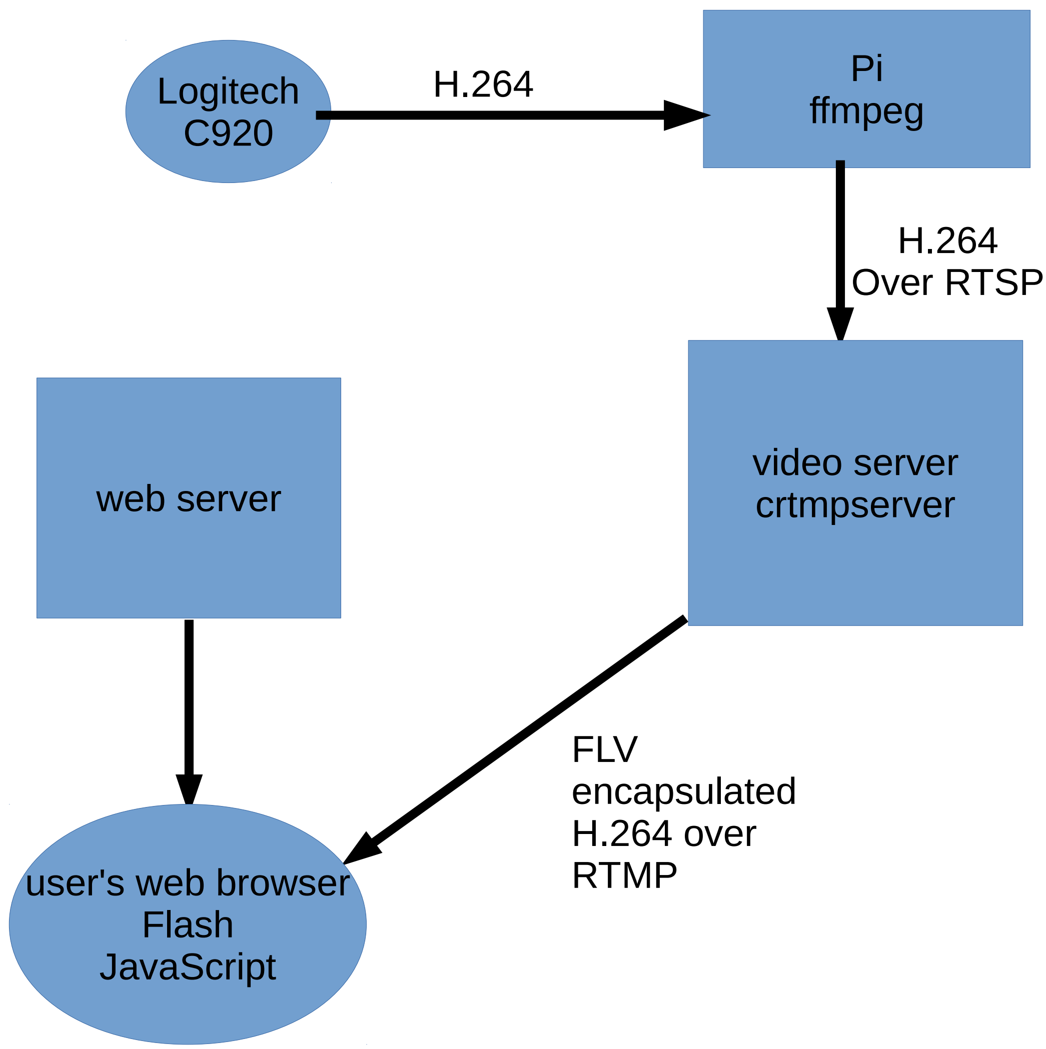


Figure 29: The method used to stream video from the pod to a web page.

For streaming the video from the pod to the web site several technologies were utilized. In the pod itself is the Logitech C920 web camera. This camera has many desirable features including a high quality lens with autofocus and image sensor. Most important to the current project is it's ability to utilize built-in hardware encoding to provide a stream of H.264 video in near real-time, even at it's maximum resolution of 1920x1080 pixels. The hardware encoding eliminates the considerable overhead and processing time associated

with encoding H.264 video on the host computer.

The Raspberry Pi computer connected to the camera therefore simply needs to receive the H.264 encoded video stream from the camera and re-transmit the data to the pier computer. To make this data transfer a Linux command-line program called ffmpeg is used. ffmpeg is desirable in that it does not necessarily have to perform any compression on it's input source before outputting it to the pier server. In a sense it is essentially acting as a Unix pipe from the raw data provided by the C920 web cam connected to the Pi to the video server using RTSP (Real Time Streaming Protocol).

The video server (or another computer at the pier) receives this RTSP stream on port 554 and encapsulates the still unaltered H.264 stream from the Pi in an FLV (Flash Video) stream. In the current implementation this encapsulation is performed by a Linux command-line program called crtmpserver. This program then streams the Flash Video on port 1935.

The web server, which may be the same or different computer from the video server, hosts a web page with an embedded JavaScript based Flash video player called JwPlayer. JWPlayer then requests the Flash Video stream from the video streaming server's port 1935. As this is a Flash video accessibility to the stream is limited to users with web browsers that support Flash. Also, JavaScript must be enabled in the browser in order to run the video player program. This video streaming implementation was heavily inspired by the works of Many Ayromlou and Derek Molloy [9].

6.4 Pier Server

Our server will be responsible for receiving requests/commands from web clients and delegating tasks to our various sub systems. It will be communicating over a variety of different protocols with various external systems including: the Pod using JSON Remote Procedure calls over a socket, the ServoPack using Modbus over a TCP connection, the clients using AJAX, and delivering video. Our system will be implemented as multiple separate processes on various pieces of hardware.

The rails server that will be the primary handler for user interaction on the web page will run on a physical server located at the pier. The Servo Daemon will run on the pier server and will receive movement commands from the rails back end it will also act as a Modbus Client that will format the commands into Modbus Packets to be sent to the Modbus Server running on the mp2600iec. The Servo control system(Modbus Server) was implemented as a simple state machine. It waits for Boolean state transition flags located in Modbus accessible memory to be set. These flags trigger a state transition to their corresponding state of execution and when complete they return back to the wait for command state. To view the code and program the mp2600iec use the motionworks iec software that can be found at the pier2pier dropbox.

To install MotionWorks 3 download the MWiec_v3.zip file from drop box. unzip it and there should be 3 more zipped files: motionworks firmware, prerequisites and MWiec Pro. Unzip the MWiec Pro folder first Then unzip the other two into the resulting folder of the first. Open the unzipped MWiec pro folder and run the setup application.

A good place to get started on learning motionworks is yaskawa's youtube channel. Here is a link to a playlist introducing the basics of motionworks: https://www.youtube.com/playlist?list=PLNAENlyEDCkzMwkIpWNwX0DeJdlVFwyB0

The communication between the web server and the servo controller is handled by a Modbus Master Program contained in the ruby files Servo_Modbus_Daemon.rb and p2pmodbus.rb.The code makes use of a ruby gem titled rmdobus that can be acquired at "https://github.com/flipback/rmodbus".

The code currently supports incremental movements, absolute movements, setting home and moving home.

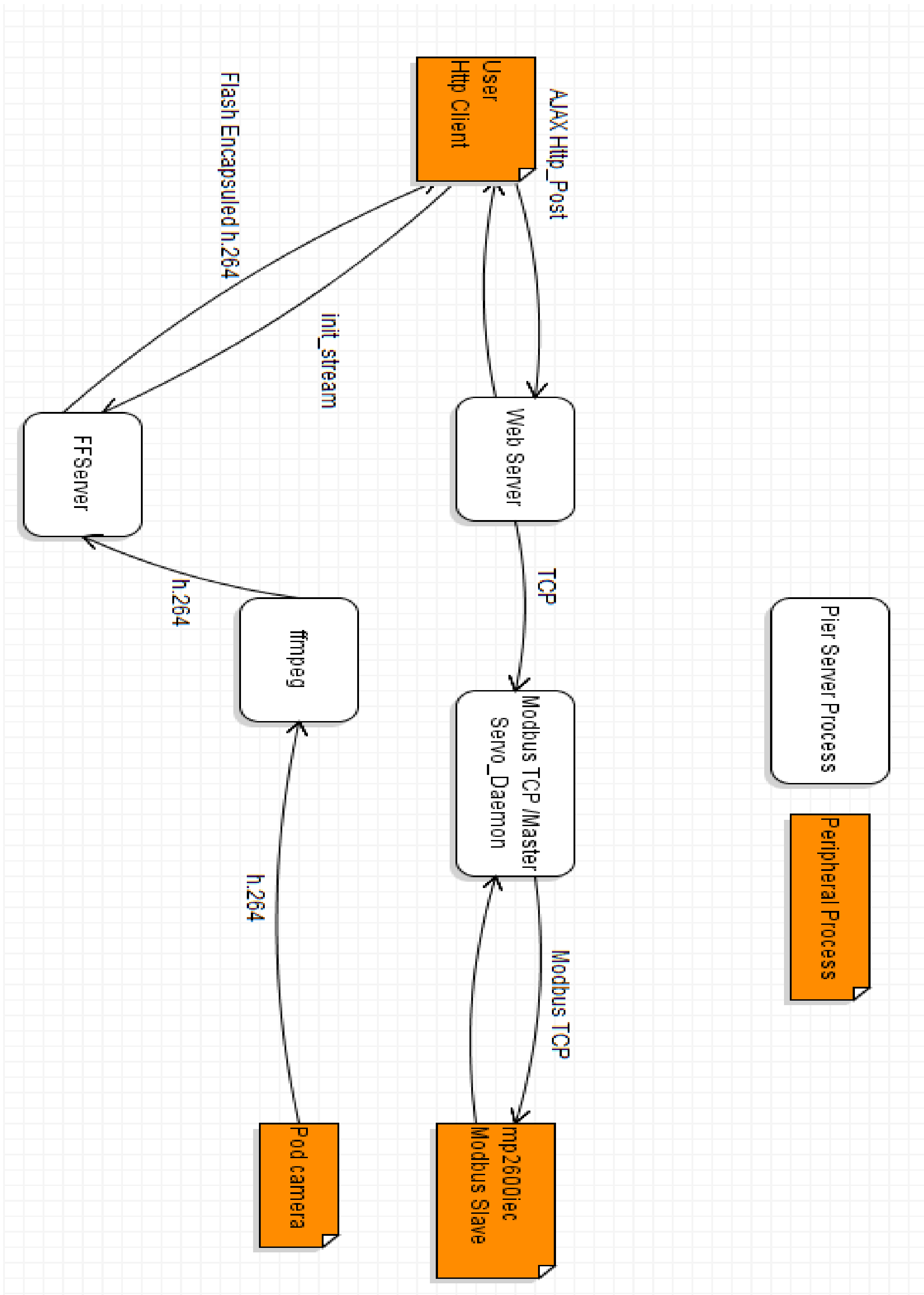


Figure 30: Shows Various Processes and their intercommunication

6.5 Motor controller

For our servo system Yaskawa donated an mp2600iec Servopack to act as our systems motor controller and amplifier. The motor controller will be implemented as a State machine. When powered it will go through an initialization routine which will power the servo, and home the pod to the top of its path. Once initialization and homing are complete, the ServoPak will wait for a command to be issued by the pier server. The ServoPak will communicate with the pier server using Modbus over TCP. This allows the server to access and manipulate data in the ServoPak's various registers and coils. There will be a writable coil that corresponds to each transition on the state diagram below.

The servo being controlled by the motor controller makes use of a holding brake since it will be moving a load vertically. This holding brake needs to be wired to the proper digital I/O control outputs so that it is actuated correctly for enabling and disabling movement. The motor controller's digital I/O for controlling the servo's brakes are located on the CN1 output terminal. This terminal utilizes sinking open collector transistors and requires 24 volts for digital actuation. The servo brakes themselves utilize 24 volts as well and are active low in that the brakes are active when no voltage is applied. In software the Motionworks libraries for motion control are already designed to properly actuate the holding brake digital outputs; however, in the Motionworks hardware configuration the servo's brake outputs need to be set so the software know which outputs to use.

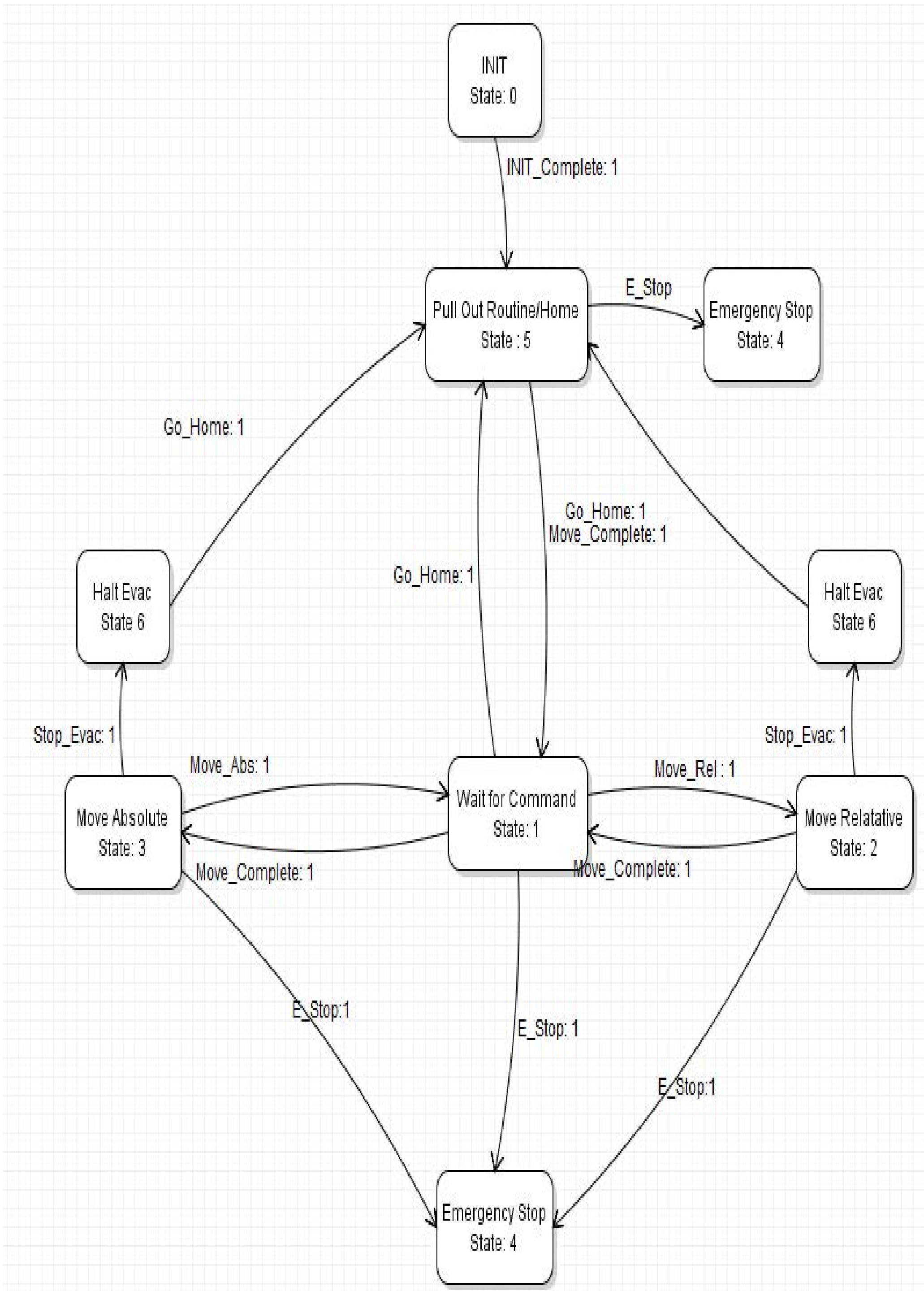


Figure 31: Describing Servo Controller implementation.

The server will communicate with the mp2600iec over an Ethernet connection using Modbus TCP. We have found an implementation in ruby that will allow us to execute all of the Modbus functionalities supported by the 2600iec. Below is a simple implementation of a Modbus client that connects to the server, executes all of the supported functions and then closes its connection.

```ruby
require 'rmodbus'
require 'benchmark'

include ModBus

ServoPack_IP = '127.0.0.1'
ServoPacl_Port = 1502

Position = 0
Count = 5
Bool_Value = 1
Value = 0xAAAA
Array_Of_Values = [0, 1, 2, 3, 4]

cl = TCPClient.new(ServoPack_IP, ServoPacl_Port)
cl.with_slave(1) do |slave|

  slave.read_coils Position, Count
  slave.read_discrete_inputs Position, Count
  slave.read_holding_registers Position, Count
  slave.read_input_registers Position, Count
  slave.write_single_coil Position, Bool_Value
  slave.write_single_register Position, Value
  slave.write_multiple_registers Position, Array_Of_Values


cl.close

```

Figure 32: A Simple Modbus client calling all supported Modbus functions (Using the Ruby gem RModBus).[found at... https://github.com/flipback/rmodbus]

6.6 Winch/Servo Connection

The winch and the servo motor have been inherited from the previous projects. We have also been provided with an aluminum enclosure box that is rated NEMA 4X to protect against the sea environment. The enclosure box will contain the servo motor and the planetary gear reduction box. The connection diagram showing how the winch and servo are connected together is shown in Figure 33, and the images of the parts are shown in Figures 34- 36.

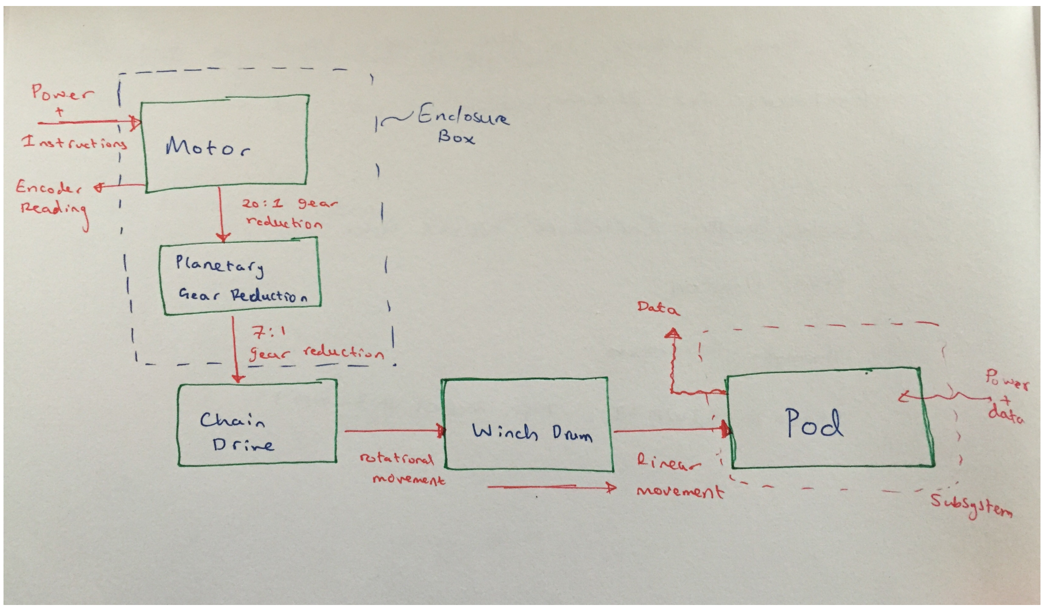


Figure 33: Flow diagram showing the connection of the mechanical components.

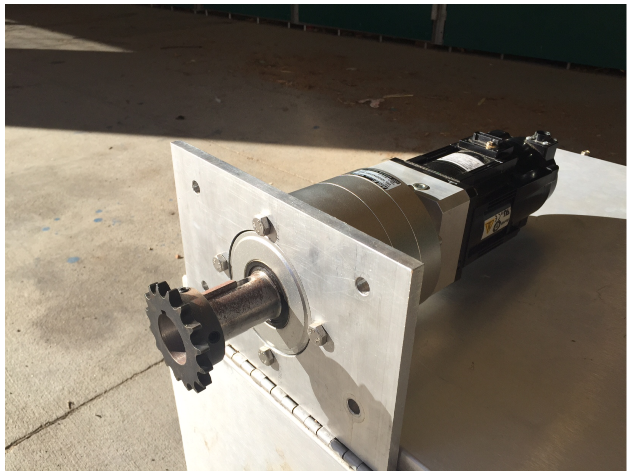


Figure 34: Image of servo motor connected to the planetary gear reduction.

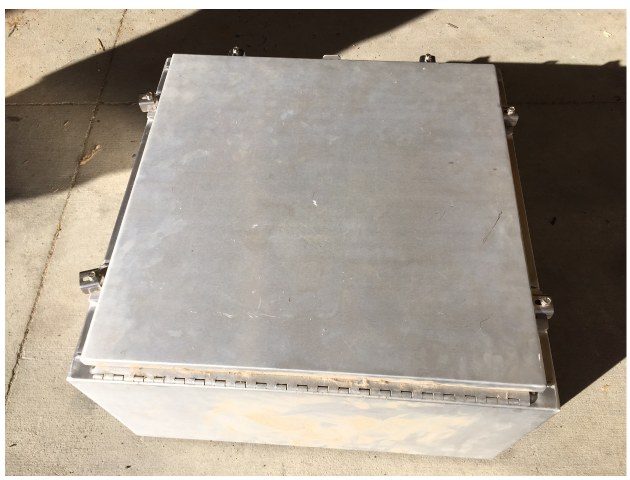


Figure 35: Image of servo motor enclosure.

Figure 36: Image of winch assembly to lower the pod.

The total gear reduction from the servo motor to the winch drum after the planetary gears and the chain drive is 140:1. The planetary gear ratio is 20:1 followed by a 7:1 gear reduction of the chain drive. The gear ratios provide a torque magnification that is large enough to lift the weight of the pod. The simple analysis can be found in Appendix G.

6.6.1 Connector Plate

The motor will be enclosed in the aluminum box, and a hole will be drilled in the enclosure for the motor shaft to pass through. The motor shaft, with the sprocket on the end will then be connected to the chain on the winch drum. The winch has holes for the motor shaft to pass through. It also has four corner holes that allow for a connector plate to be bolted to it, as shown in Figure 36.

The winch and the enclosure will be connected by a rectangular plate with a hole in the center for the shaft. This hole will be sealed by an o-ring that is placed on the shaft. A smaller plate that will hold the o-ring in place has

also been fabricated. The o-ring was chosen for its low cost. The winch and enclosure will be secured on the pier floor, and the motor shaft is small in length. Thus, there is no need for any load supporting capacity, such as that provided by a ball-bearing. The critical speed of the shaft is also well away from the operating range of the motor, refer to Appendix G. The o-ring will be tested by inspection to verify the sealing and it will be kept well lubricated to reduce the amount of friction, and also to help prevent abrasion and tearing of the o-ring. The model of the connector plate is shown in Figure 37.The 0-ring holder plate model is shown in Figure 38. The manufactured plate is shown in the product realization section.

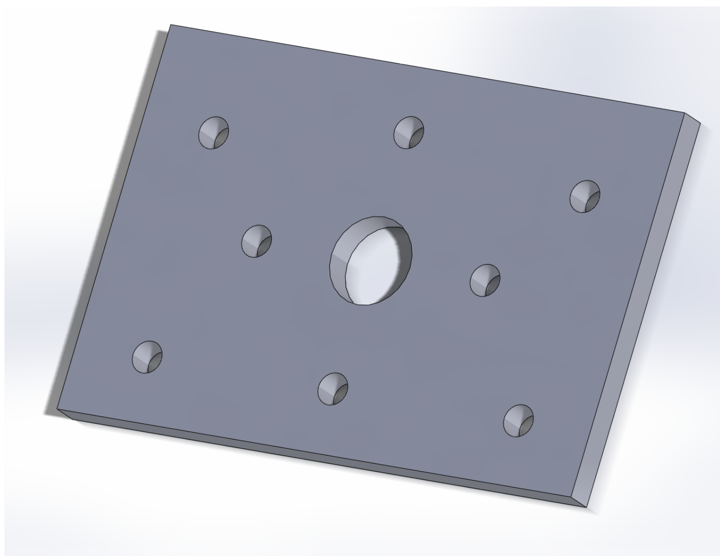


Figure 37: Connector plate model that connects the winch assembly and motor enclosure box together with holes in the middle to attach the motor.

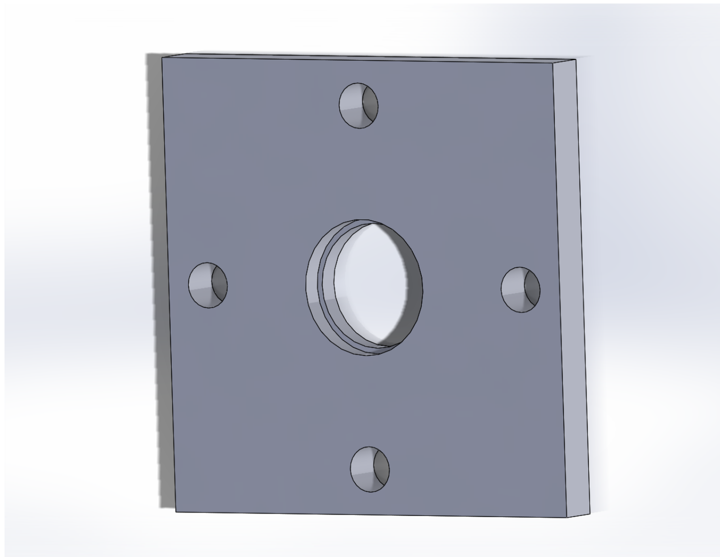


Figure 38: O-ring holder plate that will be attached to the connector plate to hold the o-ring in place when the shaft rotates.

6.6.2 Communication

Data and power signals to and from the pod will be carried by the under sea cable. The cable will be connected to the pier servers where the signals will be processed. A common way of transferring signals from a rotating frame, such as a winch, to a stationary frame, such as the servers, is by the use of a slip ring. The winch that was inherited does come with a slip ring assembly. However, sending data signals using slip rings is tricky and unreliable due the amount of noise associated with the signals.

Since the winch drum will spin for 15 rotations at most in any given direction, the use of a slip ring may not be necessary. The signals will be passed through another cable that will be spooled inside the drum and connected to the under sea cable on the winch drum. The cable will then be passed through the opening designed for the slip ring and connect to the pier server on the other end. The inside of the winch drum is shown in Figure 39.

This idea was tested using a 5 ft. long CAT-5 Ethernet cable. The cable began twisting and pinching after about 5 rotations of the winch. This shows promise for the concept, using longer cable.

This test was then repeated using a longer, 25 ft, Ethernet cable and the signal was carried without any problems. However, the concerns about reliability over the life of use were still raised due to the twisting of the cable. A solution to this problem may be to keep the cable locked in place in a loop to reduce excessive twisting. We will try this method by using tygon tubing to guide the loop of the Ethernet cable and keep it in place.

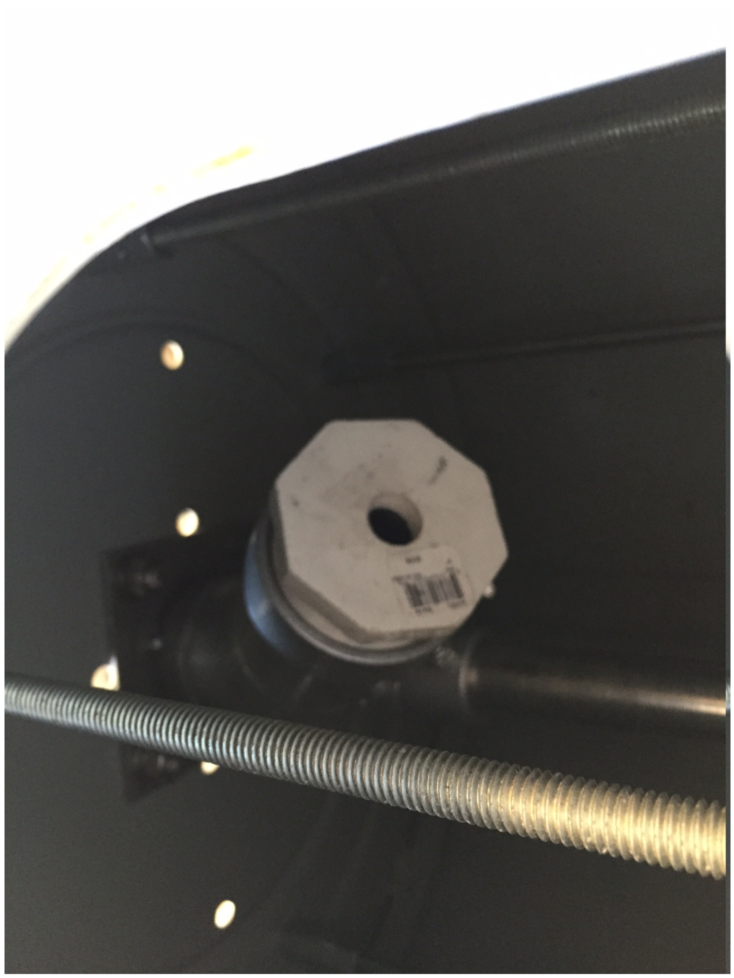


Figure 39: Image of inside of the winch drum showing the center opening designed for the slip ring.

6.7 Pod Electronics

The pod includes a number of components that were inherited from previous teams. These components include LED lights, a switch, a web camera, Raspberry Pi, Pawesome board, leak detection system, motor, a hall effect sensor, and a temperature and humidity sensor. These components communicate to the surface via a cable connector located at the top of the pod.

6.7.1 Raspberry Pi

The Raspberry Pi is responsible primarily for streaming video from the web camera inside the pod. It is connected to the pier via an Ethernet cable, which is connected to a switch in the pod, which is then connected to the pier via a waterproof data-cable.

6.7.2 Pawesome board

The Pawesome board was developed by the second team working on the pier portal project. It is a custom board responsible for a majority of the operations within the pod including; the LED lights, motor within the pod, temperature and humidity sensor, leak detection system, fans, and hall effect sensor.

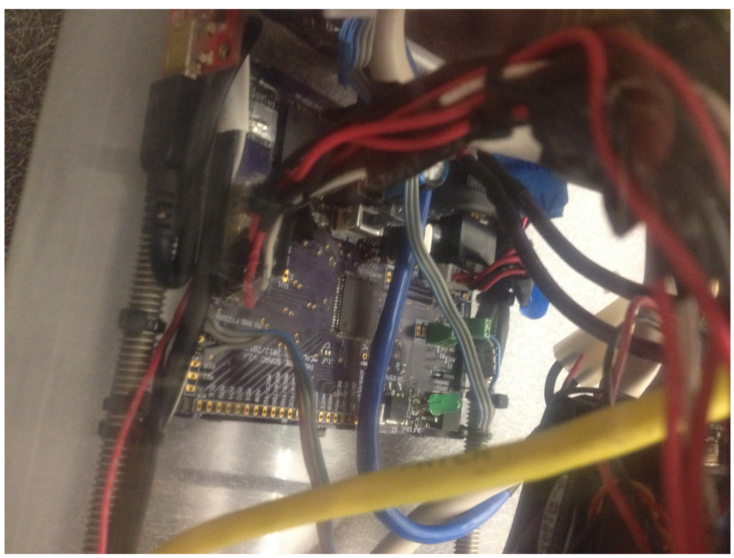


Figure 40: Pawesome board in the pod.

1. **LED lights:** The LED lights use a pair of LED lamps with 3 LEDs each. The LEDs are Cree XML warm white lights.

2. **Pod motor:** The inside of the pod uses a 24V DC motor to rotate the pod internals. This allows the camera to have a wider range of view by enabling panning.

3. **Temperature/humidity sensor:** The primary use of the temperature sensors is to ensure that the pod does not exceed the temperature requirement of 72 degrees Celsius due to the heat output from all the electronics, primarily the LED lights. If the pod does begin to overheat, then the pod is shutdown and a pullout routine is executed[27].

4. **Leak detection system:** The leak detection system is located at the bottom of the pod. It consists of a circuit which is normally low, however gets set to high if the circuit is completed by a pool of water.

5. **Fans:** Fans are used inside of the pod to distribute the heat generated by the LED lights. There are 2 fans, one located next to each LED lamp.

6. **Hall effect sensor:** The hall effect sensor currently inside the pod is used as a turn limit switch for the pod internals. A magnet is used inside the pod to determine if the motor has reached its turn limit. Another hall effect sensor will be used near the bottom of the pod in conjunction with a magnet at the bottom of the pier in order to keep track of when the pod reaches the end of the pier.

The current pod layout also has the Pawesome board located at the bottom of the pod, next to the leak detection system. We suspect that the reason for this is for easy access during development. For the final pod we will relocate the Pawesome board to the middle of the pod, along with some wiring, to avoid potential damage to the components during the event of a potential leak. Figure 41 shows the location of the leak detection system with a magenta arrow, current location of the Pawesome board with a red arrow, and the location it will be moved to with a green arrow.

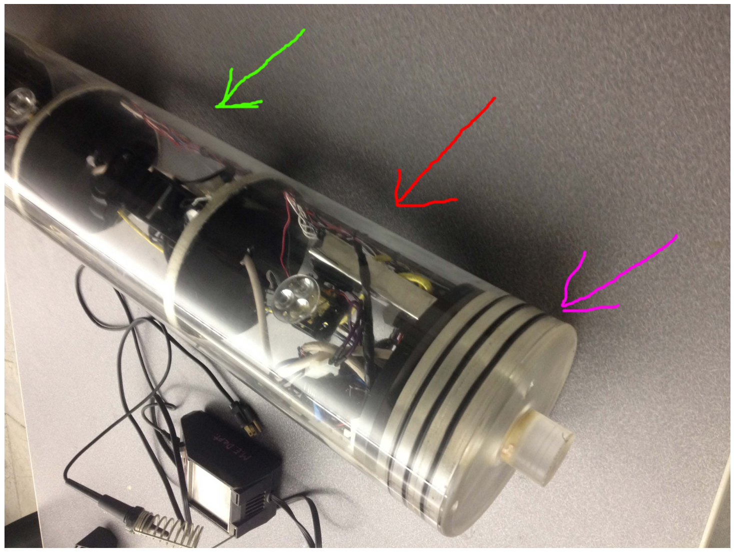


Figure 41: Pawesome board [red arrow] location relative to the leak detection system [magenta arrow] along with the final location [green arrow].

The Pawesome board will also require a case to protect it from potential leaks that occur in the pod. The case will not need to be waterproof itself, however it should prevent water from directly contacting the board.

6.7.3 Web Camera

The web camera used in this project is inherited from the previous group, a Logitech C920. It is capable of hardware video compression of H.264 high definition video. The technical specifications for the web camera recommends 1Mbps upload/download speed for 720p high definition video, which just falls within our bandwidth requirement of 1Mbps[24].

6.8 Costs

1. Winch and Motor Enclosure: $399 The Winch and Motor Enclosure is required to protect the winch and motor from the ocean environment,

which can cause corrosion and salt deposits harmful to the project's components. This is a prefabricated plastic enclosure that will be acquired from Home Depot locally.

2. Connecting Plate and O-Ring: $37.19 As discussed above, the connector plate is required for the winch-motor interface. It will be machined at the shop on campus out of aluminum purchased from McMaster-Carr* at a price of $28.52. The O-ring will also be purchased from McMaster-Carr for $8.67.

   * We learned that it may be possible to buy aluminum stock from the machine shop. If it is available for a cheaper price, than it will be bought from the machine shop.

3. Ethernet Cables: $82.48 We will purchase several different Ethernet cables to test and find the optimal cable for this project. These will all be acquired from Amazon.com and range in price from $4.20 to $27.01 each. The first cable is a fifty foot Cat5e cable. We likely will not end up using such a long cable, but it is actually cheaper than many of the shorter cables at $4.20 so we will purchase it just in case. The next cable is a twenty-five foot Cat5e cable that costs $7.99. A cable of this length will likely be used in our final design, but we are not sure if a Cat5e or Cat6 cable would suit our needs more effectively. The next cable is a twelve foot long Cat5e cable for $6.99. While shorter than the other cables selected, it is still longer than the roughly five foot long cable we used in preliminary tests that proved to be promising. Next is a twelve foot Cat6 cable for $14.50. We are interested in finding out the difference between Cat5e and Cat6 cables in this application. Therefore we have also selected a twenty-five foot Cat6 cable for $21.79 and a thirty-five foot Cat6 cable for $27.01.

4. Emergency Stop Enclosures: $34.34 The Emergency Stop Enclosures will protect the emergency shut-off switches previous teams have purchased from the ocean atmosphere. They are $17.17 a piece and will be purchased from Grainger. These enclosures are rated to withstand significant ocean environmental factors such as water exposure and corrosion.

The list of vendors for the items and the vendor supplied data sheets can be found in Appendix C and Appendix D respectively.

6.9 Safety Considerations

Great care must be taken when using a 200V power outlet. These outlets produce a very high voltage that can injure operators. No work should be done on the system when it is running. The other area of safety concern is pinch points on the winch when it is rotating. Under normal operation, the system will be enclosed in a shed. Once again, no human operators should be touching the winch or motor when the system is powered.

6.10 Maintenance and Repair

The biggest concern with the project is rusting of the metal components and accumulation of marine life on the track that would hamper the pod's movement. To prevent rust, the parts will be painted with protective paint and the motor shaft and gears will be covered with lubricant.The system should be maintained by regular monitoring from the pier staff. When signs of rust buildup appear, it should be cleaned, and the surface repainted. They should also ensure that sufficient lubrication is present where the parts are moving. With the current design, it is difficult to do a quick visual inspection to look for rust buildup. It is also difficult to reach in and clean the system easily, and it is impossible to do so when it is running. We have suggested a new design alternative in the recommendations, which allows for quick visual inspection and is expected to rust less.

With the system running constantly, marine life is expected to be discouraged from growing on the tracks and thus keeping them fairly clean. The pod will be kept clean by employing a regularly scheduled fresh water rinse system. A suggestion for the rinse system is also mentioned in the recommendations.

7 Product Realization

7.1 Manufacturing processes Employed

The only component of the project that was physically manufactured is the o-ring holder plate. The plate was constructed out of corrosion resistant 5083 Aluminum. The plate was manufactured at the Cal Poly machine shops. It was milled from a flat stock plate. The center through hole was drilled on the mill first and then a boring head was used to produce the countersink for the o-ring to sit in. Refer to Figures 45 and 46 for images of the manufactured o-ring plate.The mounting plate of the motor is also shown for reference purposes in Figures 42 - 44. The drawing of the plate can be found in Appendix B.

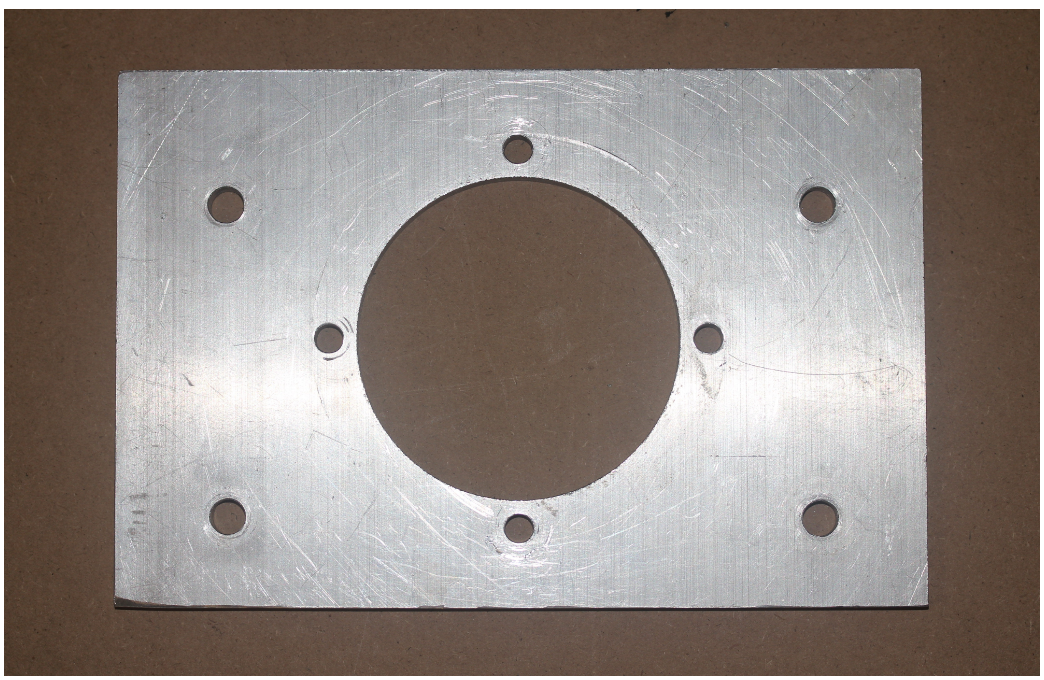


Figure 42: Front view of Motor Mount Plate.

Figure 43: Side view of Motor Mount Plate.

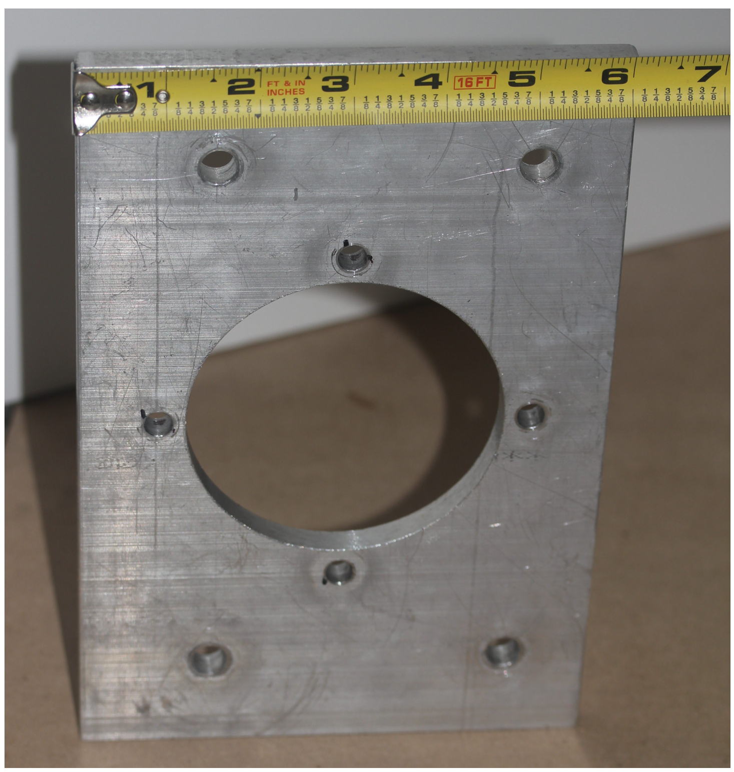


Figure 44: Front-on view of Motor Mount Plate.

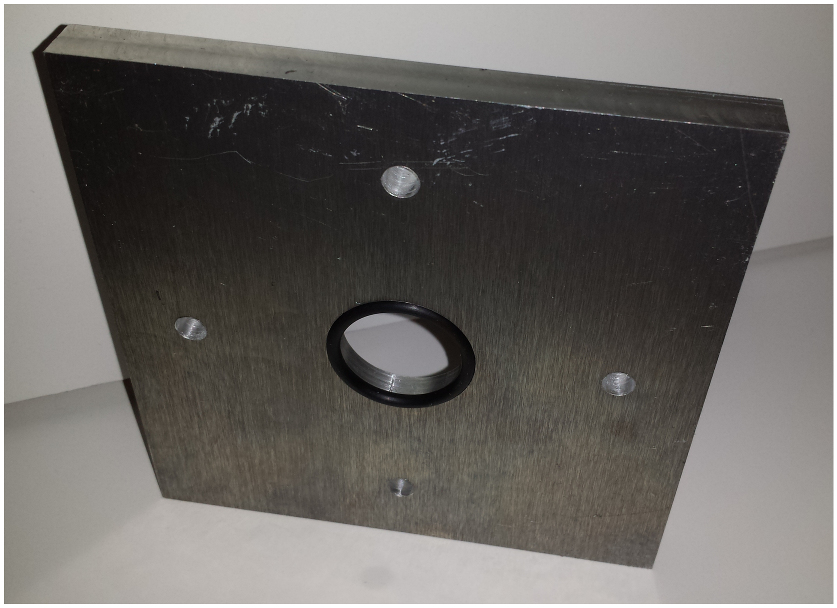


Figure 45: Profile view of O-Ring plate.

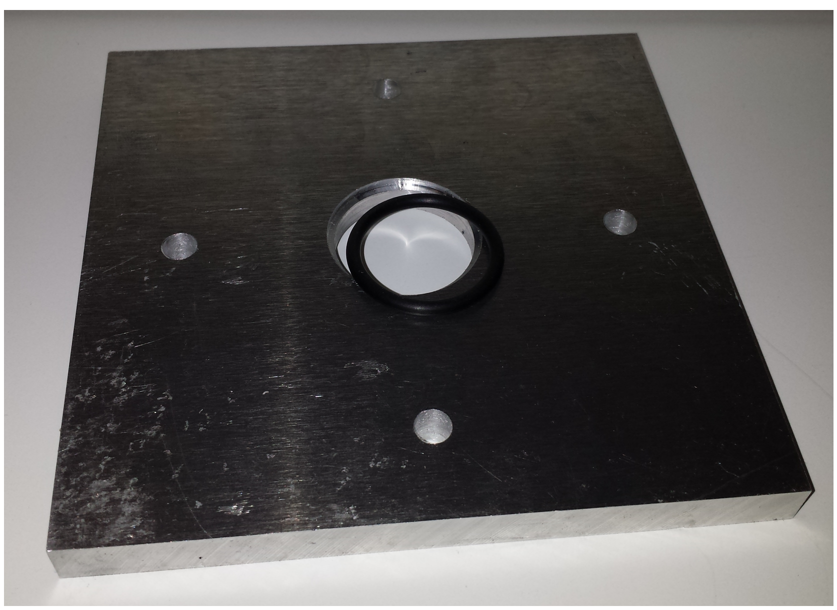


Figure 46: O-Ring plate.

7.2 Differences Between Prototype and Planned Design

Communication with the Pawesome Board was not established as discussed in the prototype. Attempts to send commands to the Pawesome board were met with an error stating a leak was detected. The reason for this behavior could not be established as the source code running on the Pawesome Board was not available for dissection. Therefore control of the pod's lights and rotational motor was not achieved.

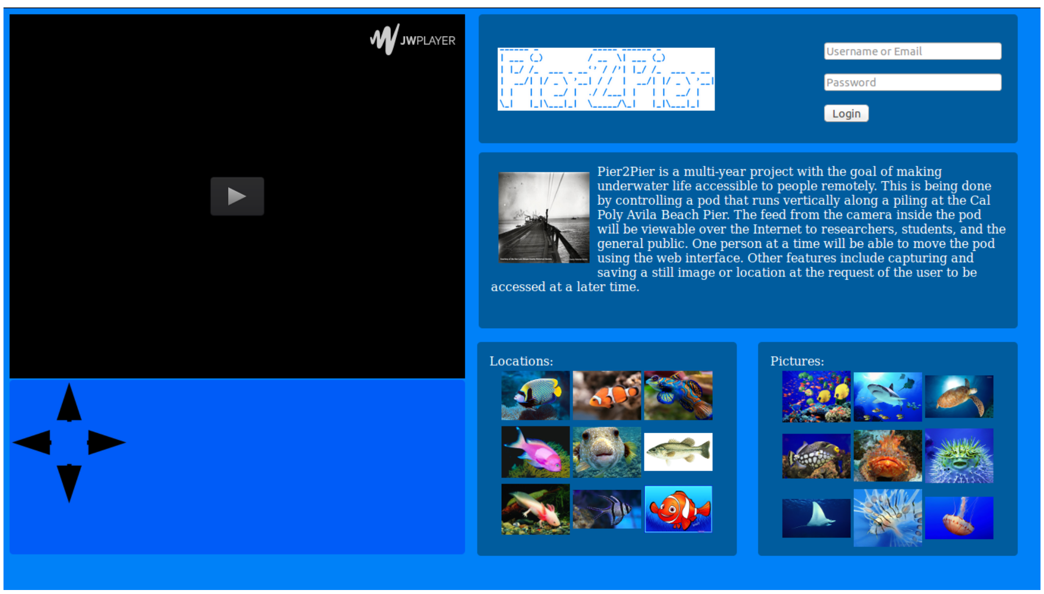


Figure 47: Screen-shot of final Web Page

The web site is lacking in some of the features specified in the final design. For instance, there is no user control system implemented. Multiple people could attempt to control the pod through the website at the same time which could lead to erratic behavior. The current team does not expect implementing such a system would be difficult; it just did not have enough time to do so. As currently implemented there is a log-in area with username and password text fields but no action is performed on submission. Following from that discrepancy, there are no user types like Researcher, Educator and General Public as specified in the final design. Implementing

these classes of users should be straightforward as well. Other things left to be implemented are a photo gallery, a way to save locations, and a video recording feature. Both the photo gallery and location saving features should be easy to implement. The current group could not determine a suitable way to allow recording videos while simultaneously streaming though there should certainly be feasible solutions. Also, there is currently no implemented scheduling system for requesting access to the pod controls at a certain time in the future.

In the mechanical system, the servo itself was not outputting enough torque to effectively move the winch drum and lift the pod. There appeared to be a mechanical torque limit in the gear box attached to the servo's pinion. We recommend taking the gearbox off of the servo to determine what causes the torque limit. Another recommendation for future groups is to find a replacement gearbox.

7.3 Recommendations for Manufacture of Design

Several issues were encountered developing on the Raspberry Pi. Some of these issues could not be resolved in the allotted time, so it was abandoned. Issues were encountered with several Raspberry Pi B+s and two Raspberry Pi 2s. It is possible some of these issues resulted from unrelated software or hardware conflicts with the Pi, but the recommendation for future iterations of this project is to consider replacing the Raspberry Pi in the project with another small computer like the Beagle Bone Black.

Another possible solution to issues with the Raspberry Pi is to replace it with a computer located on the pier. A potential problem with this approach is the length of the USB cable required for the pier computer to receive video from the camera mounted in the pod. Several powered USB extension cables would be required to implement this solution.

Still another possibility is to use an active Cat 5 Ethernet USB extender to cover the distance from the pier to the on-board camera. Then a standard full-featured desktop computer located on the pier could take the place of the Pi in streaming the video from the pod camera. In that case, the Pi

or some other similar device would likely still be needed to control the pod lights and rotational motor. There are several such products available[5][4].

7.4 Cost Estimate for Future Production

An additional $500.00 would be required to complete development of this project. This includes the cost of a Beagle Bone Black, an Ethernet cable and a housing for the motor-winch setup.

8 Design Verification

8.1 List of Tests

A list of tests was developed to be performed on the subsystems to ensure that they meet the engineering specifications listed in A.The list of tests is shown in Table 7. In addition to the tests listed in the table, we also performed an Ethernet cable test to determine the viability of not using a slip ring. Detailed test procedures can be found in Appendix F.

Table 7: Test plan and descriptions of tests for the engineering specifications.

	Report Date: 2/27/2015		Sponsor	Tom Moylan					
TEST PLAN									
Item No	Specification	Test Description	Acceptance Criteria	Test Responsibility	Test Stage	SAMPLES TESTED		TIMING	
						Quantity	Type	Start date	Finish dat
1	**Submersibility**	Submerge pod at various depths with sponge	No leaks at a depth of 13 meters for 30 minutes	Whole team	PV	1		3/31/2015	4/2/2015
2	**Data Transfer Rate**	Run network metric tests	Network usage remains at or below 1Mb/s	Tony	DV	1		4/12/2015	4/12/2015
3	**Time to Change Component**	Time component replacement for various parts	Time to change any one component does not exceed 2 hours	Chahan	PV	5		3/19/2015	3/25/2015
4	**Data Storage Capacity**	Observe reported available space by Operating System storing videos and pictures	reported available space within 10 GB of 500 GB	Kevin	DV	1		4/12/2015	4/12/2015
5	**Camera Span Range**	Rotate camera and measure field of view	field of view of at least 60 degrees	Paul	PV	1		3/15/2015	3/16/2015
6	**Video Quality**	Observe file information reported by OS for videos	reported resolution must be 1920x1080	Kevin	DV	20		3/15/2015	3/15/2015
7	**Video Feedback Delay**	Use network metric tests to measure latency	must be no more than three seconds	Kevin	DV	3		4/26/2015	4/26/2015
8	**Vertical Speed**	Measure time taken to travel a certain distance	speed must be less than .5m/s	Chahan	PV/DV	1		3/7/2015	3/7/2015
9	**Concurrent Controlling users**	Attempt to have multiple users control the pod	no more than one person will have control at one time	Tony	PV/DV	5		4/26/2015	4/28/2015
10	**System Compatibility**	Test with multiple OS's and devices	site must be usable by computers of different specifications and running Windows, Mac, and Linux	Tony	PV/DV	15		4/26/2015	5/6/2015
11	**Lighting Intensity**	Measure light intensity	intensity must be at least 2000 lumens	Paul	PV	1		3/15/2015	3/15/2015
12	**System complexity**	Count number of components	must be reasonable number of components	Kevin	DV	1		3/22/2015	3/22/2015
13	**Tension Drop Standby**	Do a pull/push test and ensure motor stops	motor must stop when it encounters a tension of 30N or more	Paul	PV	3		4/14/2015	4/16/2015
14	**Automated Freshwater Rinse**	Ensure freshwater rinse routine runs twice a day	rinse must run twice a day	Whole team	CV	1		5/2/2015	5/4/2015
15	**Reliability**	Accelerated cable lowering and raising test	must continue to communicate after 100 consecutive round trips	Paul	DV	1		3/7/2015	3/9/2015
16	**Overhead**	Measure packet size and compare to required data size	at least 70% of the packet must be required data	Tony	PV	5		5/2/2015	5/2/2015

We were not able to carry out all the tests listed in the table. We tested the video feedback delay, system and browser compatibility, and also the cable test.

8.2 Ethernet Cable Test

Ethernet cables of different lengths were wound around the winch, connecting two laptops. The durability of each cable type and length was determined by measuring the reliability of the electrical connection produced. This was measured by sending pings from one computer to the other and measuring the success rate at each number of twists in the cable. The test was successful in transmitting signals through the rotations without any packet loss in the process. However, we are concerned with the reliability of the cable over a longer period of time. The cable ends up being twisted around itself over time and does not unwind with reverse rotation. This can cause the cable to be damaged quickly.

Testing was also done by placing a Tygon tube in the winch and rotating it. The tubing exhibited the same worrying twisting behavior of the Ethernet cable, so routing the Ethernet cable through the Tygon tube would not alleviate our concerns in that regard. However, the current group would recommend encasing the Ethernet cable in a Tygon tube anyway as doing so would add a protective layer between the Ethernet cable and potentially sharp metal edges on the winch.

8.3 Latency Test

Latency was found to be roughly 2.5 seconds, when streaming video and accessing it on the same computer. When the test was done by streaming on one computer and accessing the video on another computer across a widely used wireless network latency was found to be approximately 5 seconds. Appropriate equipment was not available to test latency across a wired Ethernet connection, but is likely to be lower than the 5 second latency across the wireless network.

8.4 Cross Browser Compatibility Test

The website with video streaming was found to be compatible with all major browsers on both Linux and Windows with Flash already installed. Without Flash or with Flash disabled the video would not stream in any browser. Computers running Mac OS X where not available for testing but they should be compatible with the web site and video streaming.

9 Conclusions and Recommendations

9.1 Conclusions

This project was deceptively complex for such a seemingly straight-forward problem. The future implementation will involve considerable mechanical and electrical engineering in addition to heavily detailed and interconnected programming solutions at widely varying levels of abstraction. It was made more complex by the various stages of progress made by previous teams. Many component selection decisions of the current design were heavily influenced by other now irrelevant component selection decisions from prior project incarnations. Much of the work done on this stage of the project involved tracking down small pieces of information in the three previous reports produced for this project.

9.2 Recommendations

We recommend a more centralized repository, with version control, for extensive project documentation and code storage. This would reduce the confusion surrounding different stages of the project and exactly what was accomplished, how it was accomplished, and what was planned to be accomplished but never completed. This would also reduce the amount of time spent trying to acquire such data from reports giving just a broad overview of a given stage of the project. Such a repository would give greater insight

into why each design decision was made.

For this reason we have included all of our teams code on a public GitHub repository located at https://github.com/Pier2Pier. The code in this repository can be branched or forked by future teams, or used simply as a reference. The password for accessing the account is the same as the password needed to access the pier computer account in the mechatronics lab.

Another recommendation would be a more standards-compliant solution, if we were to take on this project from scratch. Avoiding custom made parts as much as possible means better documentation is available for each part and they are easier to replace as they become inoperable or obsolete. Standards compliant parts are also much easier to interface with other parts and can be expanded as necessary.

We also recommend exploring a different means of connecting the motor to the winch. The current design with the connector plate and the o-ring holder is difficult to implement due to the small amount of space available. It also results in having part of the speed reducing gear box out of the motor enclosure box and exposed to the elements. This has to be done in order to prevent placing the pinion off the edge on the motor shaft. A solution to this would be to have a longer shaft. However, a better solution might be to use a longer chain on the winch and placing the motor enclosure box adjacent to the winch. Then, a smaller size hole can be drilled to allow room for just the motor shaft to pass through the enclosure box with the rest of it protected. It would also allow for better sealing options.

A final recommendation for the project is to add the capability for a freshwater rinse system to be digitally actuated by the mpiec controller. There is a senior project that has been completed by the group Swim Free which is a freshwater pumping device that would be perfect for incorporating into the pier system for this purpose. We recommend using an ac relay to control the power provided to the pump system so that it powers on for a period whenever the pod is in the pumps vicinity.

10 Acknowledgments

We would like to thank our sponsor, Tom Moylan, and our advisor, Dr. John Ridgely for the support and valuable advice provided throughout the year. We would also like to acknowledge Yaskawa Electric Corporation for providing the servo motor, motor controller, and software support. We would also like to thank Dr. Westphal, the director of the fluids lab, for allowing access to lab space with a 200 V power supply and an area to test the motor controller.

References

[1] Alex Klimaj, Patrick Noble, Rudy Valdez *Pier Portal: Phase 3.* California Polytechnic State University, San Luis Obispo, California, Senior Project, 2014.

[2] Amazon.com *Belkin A3L791-12-BLU 12-Foot RJ45 CAT5E Patch Cable (Blue).* Internet: http://www.amazon.com/Belkin-A3L791-12-BLU-12-Foot-CAT5E-Patch/dp/B00080G0H4/ref=sr_1_2?s=electronics&ie=UTF8&qid=1423450941&sr=1-2&keywords=12+ft+ethernet+cable, February 2015.

[3] Amazon.com *C2G / Cables to Go 00930 Cat6 Snagless Shielded (STP) Network Patch Cable, White (35 Feet/10.66 Meters).* Internet: http://www.amazon.com/C2G-00924-Snagless-Shielded-Network/dp/B00ER7HDNE/ref=sr_1_5?s=electronics&ie=UTF8&qid=1423451259&sr=1-5&keywords=12+ft+ethernet+cat6, February 2015.

[4] Amazon.com *HDE USB over Cat5/5e/6 Extension Cable RJ45 Adapter Set: Computers.* Internet: http://www.amazon.com/HDE-over-Extension-Cable-Adapter/dp/B004XYEXX4, June 2015.

[5] Amazon.com *IOGEAR USB Ethernet Extender GUCE51 (Black): Electronics.* Internet: http://www.amazon.com/IOGEAR-Ethernet-Extender-GUCE51-Black/dp/B000O2X2OA, June 2015.

[6] Amazon.com *Mediabridge Cat5e Ethernet Patch Cable (25 Feet) - RJ45 Computer Networking Cord - Blue.* Internet: http://www.amazon.com/Mediabridge-Cat5e-Ethernet-Patch-Cable/dp/B001W28L2Y/ref=sr_1_1?s=electronics&ie=UTF8&qid=1423450893&sr=1-1&keywords=25+ft+ethernet+cable, February 2015.

[7] Andrew Belis, Andy Crafts, Jeremy DePangher, Aaron Hein, Michael Machado, Aaron Poulos, *Pier Portal.* California Polytechnic State University, San Luis Obispo, California, Senior Project, 2012.

[8] Automation.com *Yaskawa Releases MP2600iec Motion Controller.* Internet: http://www.automation.com/product-showcase/yaskawa-releases-mp2600iec-motion-controller, November 22, 2010.

[9] Ayromlou, Many *NERDlogger.com >> Blog Archive >> Streaming 1080P video using Raspberry Pi (or BeagleBone Black).* Internet: http://webcache.googleusercontent.com/search?q=cache:edgNlPiw-GMJ:nerdlogger.com/2013/11/09/streaming-1080p-video-using-raspberry-pi-or-beaglebone-black/+&cd=1&hl=en&ct=clnk&gl=us, June 6, 2015.

[10] Bretzel-gmbh.de *MP2600iec Webinar.* Internet PDF: http://www.bretzel-gmbh.de/assets/files/dokumente/posi/mp26/Product_MP2600iec_290420 August 31, 2011.

[11] Cal Poly Biological Sciences Department *Field Study Sites and Specimen Collections.* Internet: http://bio.calpoly.edu/content/field-study-sites-specimen-collections, January 2015.

[12] Centr Camera Inc. *360 degree video camera recording.* Internet: http://www.centrcam.com/, October 2014.

[13] Coder Ranch Forum *Definition of server deployment.* Internet: http://www.coderanch.com/t/475835/BEA-Weblogic/deployment-web-application, December 2014.

[14] CodeSamplez *HTML5 video streaming with PHP.* Internet: http://codesamplez.com/programming/php-html5-video-streaming-tutorial, December 2014.

[15] Columbus Zoo, *The Discovery Reef Aquarium Camera.* Internet: http://www.nationwide.com/cps/nw-your-zoo-view-aquarium.html, October 2014.

[16] Django Deployment *Deployment checklist.* Internet: https://docs.djangoproject.com/en/dev/howto/deployment/checklist/, December 2014.

[17] ExmouthCam *The Exmouth Seafront Beach Webcam.* Internet: http://www.exmouthcam.co.uk/webcam/, November 2014.

[18] Explore Annenberg, *Oceans: West Coast Sea Nettles.* Internet: http://explore.org/live-cams/player/seajelly-cam, October 2014.

[19] Github *Django video streaming.* Internet: https://github.com/andrewebdev/django-video, December 2014.

[20] Google *YouTube live streaming.* Internet: https://developers.google.com/youtube/v3/live/, December 2014.

[21] Grainger *Push Button Enclosure, NEMA 4,4X.* Internet: http://www.grainger.com/product/DAYTON-Push-Button-Enclosure-32W279?nls=0&searchQuery=32w279, February 2015.

[22] Home Depot *6 ft. 4 in. x 4 ft. 8 in. Slide-Lid Shed.* Internet: http://www.homedepot.com/p/Rubbermaid-6-ft-4-in-x-4-ft-8-in-Slide-Lid-Shed-1800005/203137956#specifications, February 2015.

[23] ITConversations *Interview with creator of PHP.* Internet: http://itc.conversationsnetwork.org/shows/detail3298.html, December 2014.

[24] Logitech *Logitech HD Pro Webcam C920 specifications.* Internet: http://www.logitech.com/en-us/product/hd-pro-webcam-c920, February 2015.

[25] Mammoth Mountain Ski Area *Main Lodge Cam.* Internet: http://www.mammothmountain.com/winter/ski-ride/mountain-information/cams/main-lodge-cam, November 2014.

[26] McMaster *Polyurethane O-Ring, Abrasion-Resistant, High-Strength, Dash Number 218.* Internet: http://www.mcmaster.com/#9558k39/=vtuiod, February 2015.

[27] Misha Balingit, Cory Spieler, Aaron Jen *Pier Portal II.* California Polytechnic State University, San Luis Obispo, California, Senior Project, 2013.

[28] MotionWorks IEC Hardware Configuration *Configuration guide for proggramming PLCs using Motionworks iec.* Internet PDF: https://www.yaskawa.com/pycprd/lookup/getdocument/jvgyvE5ZTUY_5CC1znzBofUgnN8I
tk11x1a3E2efE0bjZtamDqgZubAJvhD-GKkYa77DG8KrhmzU0l4-JV7JLfPOiDAah898qeRPnilMKHohUat6GENHUto0aBaImy9zpjiArE2a-3N1grcxlrH9XE4 March 26, 2013.

[29] PHP.net *PHP website.* Internet: http://php.net/, December 2014.

[30] PixController Incorporated, *Remote Outdoor Surveillance System.* Internet: http://www.pixcontroller.com/index.html, October 2014.

[31] PLCOpen.org *PLCOpen Website.* Internet: http://www.plcopen.org, February 2015.

[32] Princeton *Principle of Least Astonishment.* Internet: https://www.princeton.edu/ achaney/tmve/wiki100k/docs/Principle_of_least_astonishment.ht December 2014.

[33] Python.org *PEP 20 - The Zen of Python.* Internet: https://www.python.org/dev/peps/pep-0020/, December 2014.

[34] Python.org *Python website.* Internet: https://www.python.org/, December 2014.

[35] blog.robotip.com *Ethernet Idustrial Protocol vs Modbus TCP.* Internet blog: http://blog.robotiq.com/bid/52756/what-is-the-difference-between-ethernet-ip-and-tcp-ip, February 2015.

[36] Real Time Automation *Real Time Automation Overviews of Communication Protocols.* Internet: http://www.rtaautomation.com/technologies/, February 2015.

[37] Ruby on Rails *Getting started with Ruby on Rails.* Internet: http://guides.rubyonrails.org/getting_started.html#what-is-rails-questionmark, December 2014.

[38] Ruby-lang.org *Ruby website.* Internet: https://www.ruby-lang.org/en/, December 2014.

[39] SciPy *Python used in scientific computing.* Internet: http://docs.scipy.org/doc/scipy-dev/reference/index.html, December 2014.

[40] Simonton Court *Live Interactive Web Cam at Mallory Square in Key West FL.* Internet: http://www.simontoncourt.com/webcams/webcam_3.html, November 2014.

[41] StackOverFlow *Dedicated server for media streaming with Django.* Internet: http://stackoverflow.com/questions/9162656/stream-videos-from-django-application, December 2014.

[42] StackOverFlow *What is your preferred php deployment strategy.* Internet: http://stackoverflow.com/questions/425692/what-is-your-preferred-php-deployment-strategy, December 2014.

[43] TrendHunter Tech, *Olympic Gravity Drop Diving Camera.* Internet: http://www.trendhunter.com/trends/olympic-gravity-drop-divecam-technology-diving-camera-that-plunges-with-div, October 2014.

[44] UNESCO *Facts and figures on marine biodiversity.* Internet: http://www.unesco.org/new/en/natural-sciences/ioc-oceans/priority-areas/rio-20-ocean/blueprint-for-the-future-we-want/marine-biodiversity/facts-and-figures-on-marine-biodiversity/, December 2014.

[45] W3Techs *Usage of server-side programming languages for websites.* Internet: http://w3techs.com/technologies/overview/programming_language/all, December 2014.

[46] Webfaction *Configuring PHP.* Internet: http://docs.webfaction.com/software/php.html, December 2014.

[47] Wikipedia *Comparison of web frameworks.* Internet: http://en.wikipedia.org/wiki/Comparison_of_web_application_frameworks, December 2014.

[48] ZDNet *Desktop vs. Browser - when to deploy applications for each.* Internet: http://www.zdnet.com/article/desktop-vs-browser-when-to-deploy-applications-for-each/, November 2014.

A QFD and Engineering Requirements

Table A.1: QFD for the user requirements and engineering specifications.

Row #	Weight Chart	Relative Weight	Tom Moylan	Elementary School Teacher			Maximum Relationship	WHAT: Customer Requirements (explicit & implicit) / HOW: Engineering Specifications	Submersibility	Data transfer Rate	Time to Change Component	Data Storage Capacity	Camera Span Range	Video Quality	Video feedback Delay	Vertical Speed	Concurrent Controlling Users	System Compatibility	Lighting Intensity	System Complexity	Tension Drop Standby	Automated Freshwater Rinse	Reliability	Overhead
								Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
								Direction of Improvement																
1		6%	10	1			9	Does not Leak	●															
2		12%	5	10			9	Easy to Use							●			○					▽	
3		5%	8	1			9	Resistant to Salt Exposure	○													●		
4		3%	4	1			3	Prolonged UV Exposure	▽														○	
5		5%	8	1			9	Easy to Clean												●		○		
6		12%	10	7			9	Remotely Accessible		●								▽						○
7		8%	6	5			9	Precise and Accurate Positioning								○	●			○	▽			
8		4%	5	1			9	Modularity			●													
9		11%	7	8			9	Quality Underwater Visibility		○			○	●	▽				○			▽		
10		6%	10	1			3	Safe to Use															○	
11		9%	8	5			9	Smooth Motion								●	○			○	●			○
12		6%	3	5			9	Account Management									●	▽		○				
13		5%	2	4			9	General Use by Public		▽		▽	○	○		▽		●		▽			▽	▽
14		6%	6	3			9	Create and Schedule Subroutines				●					▽			○				
15		0%																						
16		0%																						
								HOW MUCH: Target	11 Meters	1 Mb/s	2 hours	500 GB	60 Degrees	1080 p	3 seconds	0.5 m/s	1 user	Mac, Windows, Android	2000 Lumens	< # components	< 30 N	Twice a Day	99.7 % Uptime	< 30% overhead
								Max Relationship	9	9	9	9	3	9	9	9	9	9	3	9	9	9	3	3
								Technical Importance Rating	76.39	147.3	32.95	59.68	48.64	116.7	120.7	110.4	162.2	98.58	34.05	140.9	89.56	74.85	45.5	68.17
								Relative Weight	5%	10%	2%	4%	3%	8%	8%	8%	11%	7%	2%	10%	6%	5%	3%	5%
								Weight Chart																

WHO: Customers

QFD: House of Quality

Projec Pier2Pier

Revisi 0

Date: 10/23/2014

Correlations	
Positive	+
Negative	–
No Correlation	

Relationships	
Strong	●
Moderate	○
Weak	▽

Direction of Improvement	
Maximize	▲
Target	◇
Minimize	▼

Table A.2: Decision Matrix generated for the four subsystems of interest. The top idea in each subsystem is shown in green.

Pier2Pier		Engineering Requirements													
System Functions	Potential Solutions	Data Transfer Rate	Time to change component	Data storage capacity	Video quality	Video feedback delay	Concurrent controlling users	System Compatibility	System Complexity	Overhead	Reliability	Sum +	Sum -	Sum S	Total
Specification Weight		10	9	8	9	10	8	7	15	5	19				
User Interface	Web Interface	S	S		S	S		S	S		S	0	0	79	24
	Mobile App	-	-		-	-		-	-		+	19	60	0	-41
	Desktop App	S	-		S	S		-	-		+	19	31	29	-3
Motor to controller interface	OPC		S					S	S	S	S	0	0	55	17
	Ethernet IP		+					+	+	-	+	50	5	0	45
	Modbus TCP		+					+	+	S	+	50	0	5	52
	Analog/Pulse Train		-					S	-	+	-	5	43	7	-36
Pod Electronics Control	Pawesome Board	S				S			S		S	0	0	54	16
	TI Microcontroller	-				S			-		-	0	44	10	-41
	Raspberry Pi	+				S			-		-	10	34	10	-21
	Arduino	-				S			-		-	0	44	10	-41
Web back-end	PHP	S	S	S	S	S	S		S		S	0	0	88	26
	Python	S	+	S	S	+	S		S		S	19	0	69	40
	Ruby	S	+	S	S	+	S		+		+	53	0	35	64

System	Datum
User interface	Web Interface
Motor controller interface	OPC
Pod Electronics	Pawsome Board
Web Back-end	PHP

B Final drawings

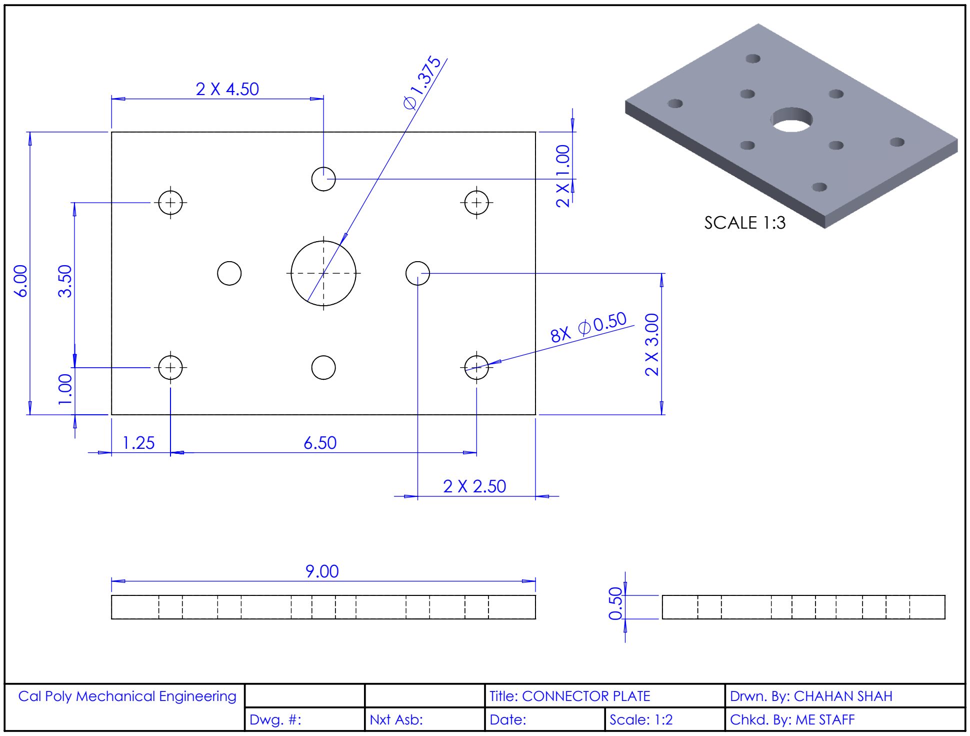


Figure B.1: Drawing of connector plate for the winch and motor enclosure.

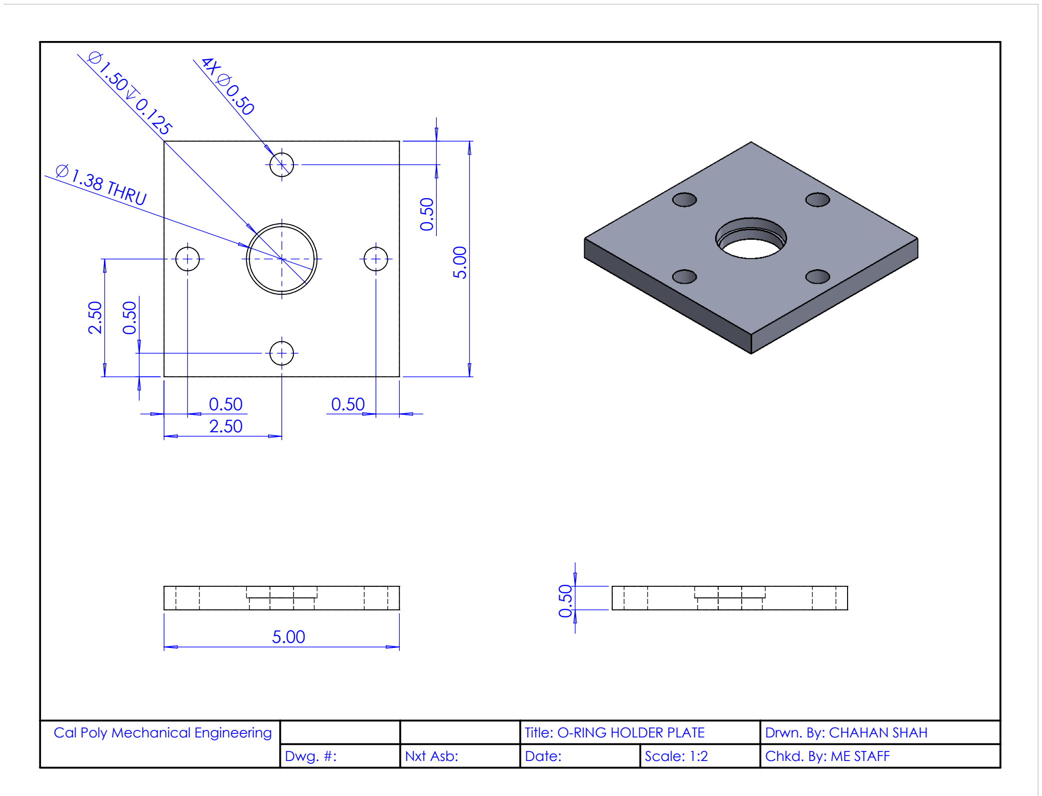


Figure B.2: Drawing of o-ring holder plate for the connector plate and shaft.

The following parts do not need to be manufactured. We already have them in our possession.They only requiring the drilling of a few holes for the bolts.

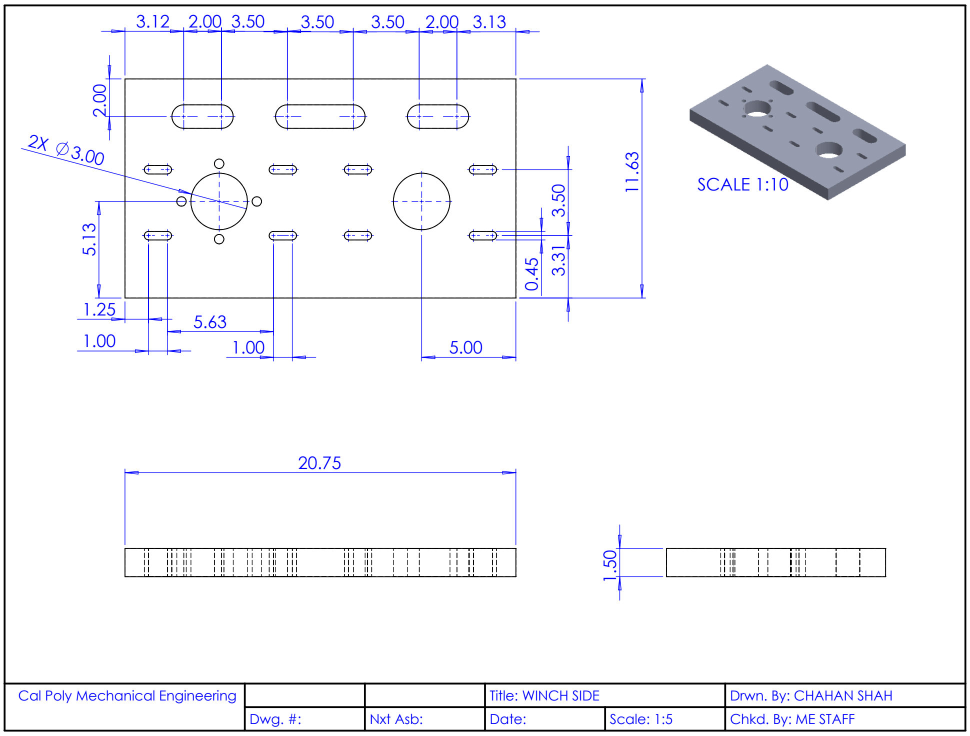


Figure B.3: Drawing of side plate of the winch that will be bolted to the connector plate.

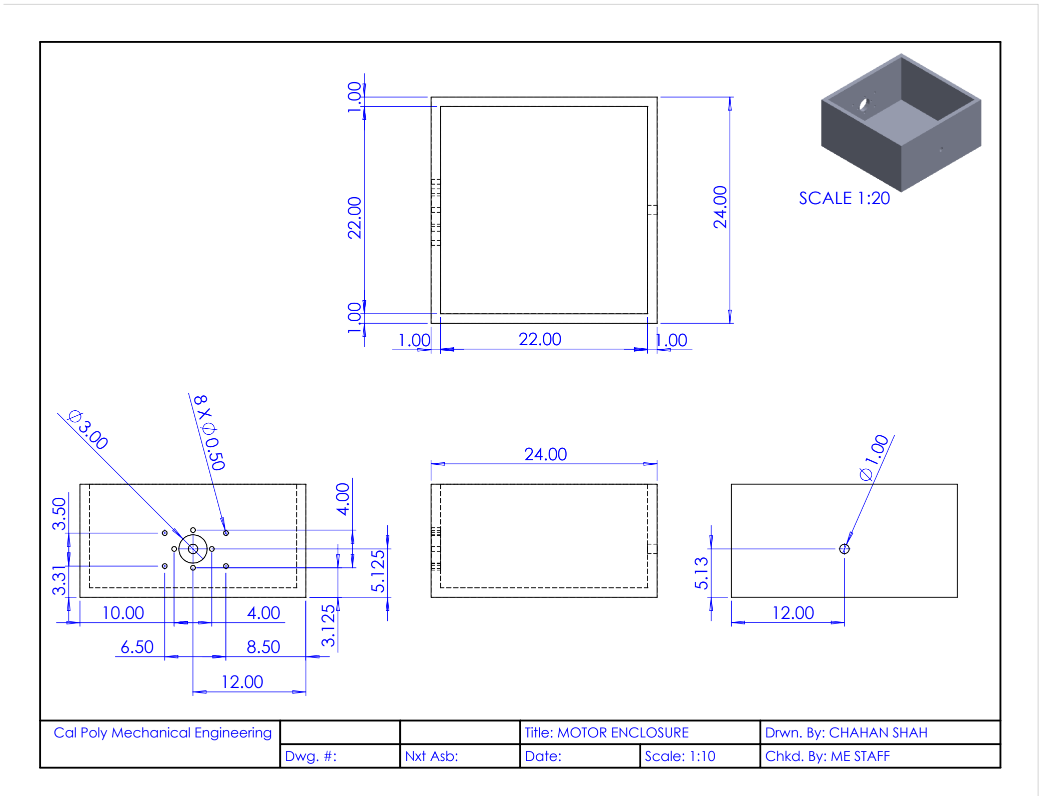


Figure B.4: Drawing of motor enclosure box.

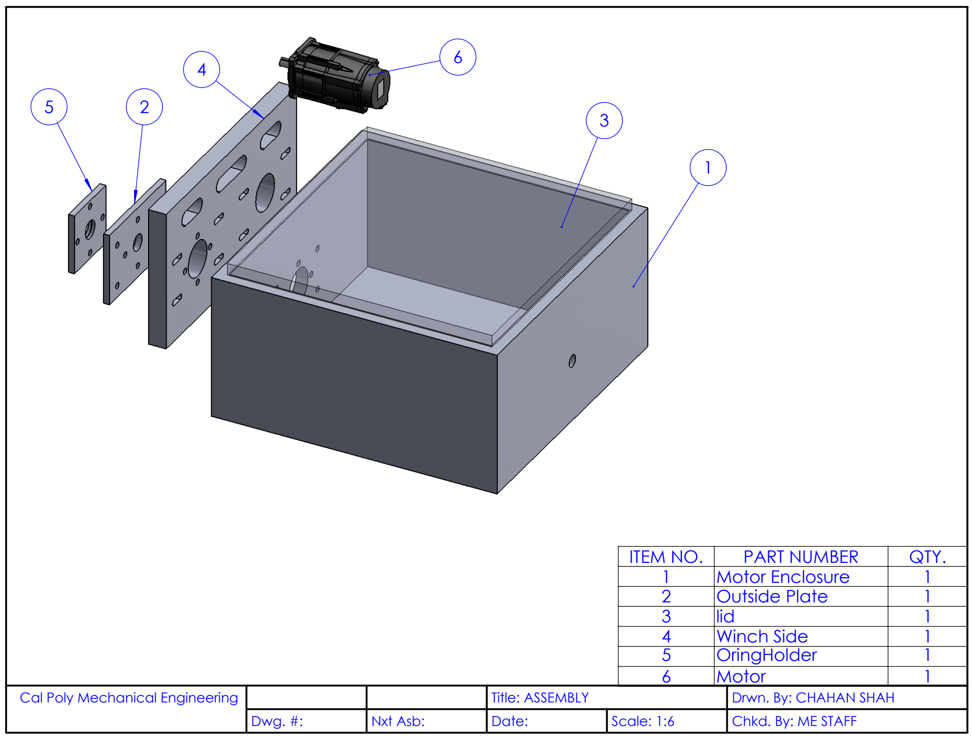


Figure B.5: Simple drawing of the assembly showing how the different pieces will connect together.

C List of vendors

Table C.1: List of vendors for all the parts to be purchased

Part	Vendor	Part Number	Total Cost	Contact Info	Delivery Time
Shed	Home Depot	814759	$430.92	1-800-466-3337	3 Days
Aluminum Plate*	McMaster Carr	8975K442	$30.80	(562) 692-5911	5 Days
O-Ring	McMaster Carr	9558K39	$9.36	(562) 692-5911	5 Days
E-Stop Enclosure	Grainger	32W729	$58	1-800-472-4643	5 Days
12ft Cat5e Cable	Amazon.com	A3L791-12-BLU	$6.99	1 (888) 280-3321	3 Days
25ft Cat5e Cable	Amazon.com	31-399-25B	$7.99	1 (888) 280-3321	3 Days
50ft Cat5e Cable	Amazon.com	0877083042452	$4.20	1 (888) 280-3321	3 Days
12ft Cat6 Cable	Amazon.com	00924	$14.50	1 (888) 280-3321	3 Days
25ft Cat6 Cable	Amazon.com	00928	$21.79	1 (888) 280-3321	3 Days
35ft Cat6 Cable	Amazon.com	00930	$27.01	1 (888) 280-3321	3 Days

*Aluminum stock plate may be available for purchase from the machine shop on campus for cheaper. If that is the case, it will be purchased from the shop.

D Vendor supplied component specifications and data sheets

Technical Specs

Item	**Push Button Enclosure**	NEMA Rating	**4, 4X**
Number of Holes	**1**	For Use With	**RP2B Operators and Plastic 22 Millimeters Pushbutton or Pilot Light Head**
Hole Size	**22.5mm**	Color	**Black Base and Gray Lid**
Height (In.)	**2.1**	Material of Construction	**Plastic**
Width (In.)	**2.68**	Standards	**IEC,UL,CSA**
Depth (In.)	**2.91**		

Figure D.1: Emergency Stop Button Enclosure Specifications[21].

SPECIFICATIONS

DIMENSIONS

Approximate Shed Depth (ft.)	6	Door Opening Width (In.)	49
Approximate Shed Width (ft.)	5	Door Opening Width (ft.)	4
Assembled Depth (in.)	76.25 in	Product Depth (in.)	76
Assembled Height (in.)	52.75 in	Product Height (in.)	52
Assembled Width (in.)	56.75 in	Product Width (in.)	55
Door Opening Height (In.)	51	Sidewall Height (in.)	44

DETAILS

Assembly Required	Yes	Roof Color Family	Black
Floor Options	Yes	Roof Material	Resin
Floor Options	With Floor	Roof Pitch	1.5:12
Foundation Included	Yes	Shed Door Type	Double
Lockable Door/Gate Latch	Yes	Shed Features	Single Door
Number of Doors	2	Shelving Included	No
Number of Windows	0	Siding Color Family	Beige/bisque
Product Weight (lb.)	177 lb	Storage Capacity (cu. ft.)	96
Returnable	90-Day		

Figure D.2: Winch and Motor Enclosure Specifications[22].

AS568A Dash No.	218
Fractional Inch Size	
ID	1 1/4"
OD	1 1/2"
Actual Inch Size	
ID	1.234"
OD	1.512"
Additional Specifications	Polyurethane O-Rings Width: 1/8" Fractional (0.139" Actual) O-Ring Sizing Chart
RoHS	Compliant

O-rings stand up to abrasion and tearing.

Polyurethane–Temperature range is -20° to +180° F. Durometer hardness is A70. Color is black.

Hard Polyurethane–Temperature range is -60° to +212° F. Durometer hardness is A90. Color is translucent to amber.

Figure D.3: O-ring specifications[26].

Cable type	Stranded SSTP
Wiring scheme	568B
Temperature rating	75DegreeC
Testing	ANSI/TIA 568 C.2 Cat6 component tested
Conductor (4 pair)	Conductor type: 26AWG (7/0.16) (bare copper)
Insulation	FM-PE
Overall cable	Foil shield: Aluminum foil (100% coverage)
Braided shield	Copper braid (40% coverage)
Jacket	PVC
Connector	Connector Type: RJ-45 Male
Shell	Nickel plated for shielding
Housing	Polycarbonate, UL94V-0
Contacts	50m gold plated
Hood	Molded Snagless PVC UL94V-0

Figure D.4: Cat 6 12, 25, and 35ft Cable specifications[3].

Product Features

- Connects network components in a wired LAN at data transmission speeds up to 1 Gbps
- 24 AWG cable exceeds TIA/EIA standards & 350 MHz bandwidth
- 50-micron male RJ45 connectors per end, with corrosion-resistant gold-plating
- 4PR UTP copper strands for minimal noise, interference & crosstalk
- Tough, yet flexible PVC jacket & heavy-duty, snag-less mold

Figure D.5: Cat 5e 25 Cable specifications[6].

Network Cable Type	Patch Cable
Network Cable Left Connector Gender	Male
Right Connector Gender	Male
Left Connector Type	RJ-45
Right Connector Type	RJ-45
Length	12ft
Enclosure Color	blue

Figure D.6: Cat 5e 12ft Cable specifications[2].

E FMEA: Failure Mode and Effects Analysis

FAILURE MODE AND EFFECTS ANALYSIS

Item:	Pod	Responsibility:	Core Team:
Model:	Current	Prepared by:	Core Team:
Core Team:	P. Maalouf, K. Nelson, C. Shah, T Miller		
FMEA number:	123456		
Page :	1 of 1		
FMEA Date (Orig	2/22/2015	Rev:	1

Process Function	Potential Failure Mode	Potential Effect(s) of Failure	Sev	Class	Potential Cause(s)/ Mechanism(s) of Failure	Occur	Current Process Controls	Detec	RPN	Recommended Action(s)	Responsibility and Target Completion Date	Action Results				
												Actions Taken	Sev	Occ	Det	RPN
Rotation	*Over Rotation*	*Damage Wiring*	*5*		*Improper hall effect sensor allignment*	*3*	*Assembly training and instructions*	*3*	*45*							
Water Tight	*Flooding*	*Damaged Electronic (Shorts)*	*8*		*Improper pod assembly*	*5*	*Operator training and instructions*	*3*	*120*	*Provide training with detailed user manual*	*5/26/2015*	-				
			8		*Pod Collision*	*3*	*None*	*5*	*120*	*Prelimary Track Inspection (After > 3 days of Not running)*		-	*8*	*2*	*2*	*32*
			8		*Leak Detection Failure*	*2*	*Leak Circuit Testing*	*8*	*128*	*Make Redundant with second Ckt*	*5/5/2015*	-	*8*	*1*	*2*	*16*
LED Function	No Longer Producing light	Bulb burns out	4		Time Wear/Tear	2		2	16							
			4		OverVoltage	1		2	8							
			4		OverHeat	4		2	32							
Fan(Cooling)	Unable to Spin	Electronics OverHeat	7		FeedBack Thermistor Fail	4		2	56	Routinely monitor Thermistor feedback, If anomoly arises Withdraw pod.	5/5/2015		5	4	1	20
			4		Wire Caught in fan	1		2	8							

Figure E.1: FMEA for the Pod Subsystem.

FAILURE MODE AND EFFECTS ANALYSIS

Item:	Server Software	Responsibility:	Core Team	FMEA numbe	123456
Model:	Current	Prepared by:	Core Team	Page :	1 of 1
Core Team:	P. Maalouf, K. Nelson, C. Shah, T Miller			FMEA Date (	2/22/2015 Rev: 1

Process Function	Potential Failure Mode	Potential Effect(s) of Failure	Severity	Class	Potential Cause(s)/ Mechanism(s) of Failure	Occurrence	Current Process Controls	Detection	RPN	Recommended Action(s)	Responsibility and Target Completion Date	Action Results				
												Actions Taken	Sev	Occ	Det	RPN
Handle databases alteration	*dataBase corruption*	*Loss of data*	*7*		*SQL Injection*	*3*	*Input sanitation*	*3*	*63*	*Improve input sanitation*	*5/26/2015*		*7*	2	*2*	*28*
Communicate commands to Servo System	*Command Not Executed*	*Pod Does Not Move as user desires*	*3*		*Movement Buffer Is full*	*5*		*1*	*15*	*Increase buffer size.*	4/5/2015		*3*	*5*	*1*	*15*
			3		Modbus Packet Dropper	3	None	7	63	Add acknowledgement packets to verify connection.	4/12/2015		3	3	2	18
Communicate commands to Pod System	Unable to locate Pod On Network	No longer able to command Pod/ Stream Video	7		Pod Connection Damaged	3	None	8	168	Add acknowledgement packets to verify connection.	4/19/2015		7	3	2	42
Single User Control	Multiple User gain Control	Pod Acts erratic relative to user perspective	5		Simultaneous login by same user from seperate terminals	4	None	7	140	Use a secure connection to allow access one at a time.	4/26/2015		5	4	1	20
			5		Authority Incorrectly given to multiple users	2		5	50	Drop sessions for all users using the pod	4/26/2015		5	2	2	20
Video Stream	Err(453) Not enough bandwidth		2		Too much simultaneous pier activity	6	Video Compression	2	24	Improve compression	5/26/2015		2	3	2	12
	Err(250) Low on Storage Space	Video Incorrectly recorded / lost	2		multiple record requests	3		2	12							

Figure E.2: FMEA for the Software Subsystem.

					FAILURE MODE AND EFFECTS ANALYSIS											
Item:	Servo/Winch System				Responsibility:		Core Team:					FMEA numbe	123456			
Model:	Current				Prepared by:		Core Team:					Page :	1 of 1			
Core Team:	P. Maalouf, K. Nelson, C. Shah, T Miller											FMEA Date (	2/22/2015	Rev:	1	
													Action Results			
Process Function	Potential Failure Mode	Potential Effect(s) of Failure	Severity	Class	Potential Cause(s)/ Mechanism(s) of Failure	Occurrence	Current Process Controls	Detection	RPN	Recommended Action(s)	Responsibility and Target Completion Date	Actions Taken	Sev	Occ	Det	RPN
Raise/Lower Pod	*Pod Falls*	*Pod Damage, Personal Injury*	*8*		*Chain Breaks*	*2*		*3*	*48*	*Maintenance (greasing / lubrication)*			*8*	*1*	*2*	*16*
			8		*Sheave Assembly Damage*	*2*		*4*	*64*	*Regular inspection of bolts and connections*			*8*	*1*	*2*	*16*
	Pod Freezes/Jams	*Excess force on motor / overheat*	*6*		*Improper machine set up*	*2*	*Operator training and instructions*	*3*	*36*							
			6		*Sheave Assembly Damage*	*2*		*4*	*48*	*Regular inspection of bolts and connections*			*6*	*1*	*2*	*12*
			6		*Animal/ Human pulling on Pod*	*3*		*6*	*108*	*Torque monitoring and system shut down*	*4/1/2015*		*6*	*3*	*2*	*36*
Home Pod	*Encoder Error*	*Forgets Home(OverShoots)*	*5*		*Encoder Battery Dies*	*5*	*None*	*9*	*225*	*Replace Battery after X*		*3/1/2008*	*5*	*5*	*1*	*25*

Figure E.3: FMEA for the Servo-Winch Subsystem.

F Test Procedures

F.1 Cable Test

Determine:
The viability of using simple cables wound around the winch drum to communicate data signals to and from the pod, thus eliminating the need for a slip ring.

Materials:

- RaspberryPi (Pi)
- External Pi power supply
- Network Equipment:
- 25 foot Cat-5 Ethernet straight through cable
- Winch Assembly
- Laptop
- Duct tape

Safety:

- Wear appropriately sized safety gloves to avoid hands and fingers being pinched by the Ethernet cable while twisting it.
- Do not reach under the winch while rotating it; only touch winch at top.
- Only one person should be rotating the winch at once. Do not get within arm's length of winch while another person is rotating it.

Procedure:

1. Duct-tape Pi to the outside of the drum. Make sure it is secure.
2. Connect Pi with an external power pack. Duct Tape the external power supply next to the Pi. Make sure it is secure.
3. Turn on the power to the Pi and PC, allow time for full initialization procedure.
4. Connect PC to the 25 foot Ethernet cable.
5. Connect the other end of the 25 foot Ethernet cable to the pi.
6. Ensure Pi and PC can communicate over the Ethernet cable by ensuring they are configured to access the same network.
7. Open the Command Prompt or Terminal, depending on the Operating System.
8. Type the following command: ping $\langle IPaddress \rangle$
9. Verify the Pi's response.
   (a) This test is deemed successful if after pinging the Pi with five packets three separate times, 100% of the packets in each set of five packets return to the PC successfully.
   (b) A ping is successful if it results in an ICMP packet being received with a type field of 0 and code of 0.
10. Disconnect the Ethernet crossover cable from the PC and Pi (Or the straight through cable from the switch and Pi) and run the end of the cable through the side opening in the winch drum and wind the cable inside of the winch drum. Pass the other end of the cable through the cut-out opening in the winch drum face and connect it to the Pi.
11. Repeat Steps 9-11 to ensure connectivity between the Pi and PC.
12. Rotate winch fifteen complete rotations, and repeat steps 10 and 11 through each rotation to ensure connection is maintained throughout.

F.2 Data Transfer Rate Test

Determine:
Verify the advertised 1 MBps network speed at the pier to ensure smooth operation of video and photo streaming.

Materials:

- laptop
- Cat5e Ethernet Cable

Procedure:

1. Connect Laptop to switch at pier using Ethernet Cable.
2. Once authenticated on the network, access the website http://www.speedtest.net.
3. Run the speed test on the website five times.
4. Average the reported upload speeds of the five runs.
5. Verify the average upload rate is at least 1 MBps.

F.3 Vertical Speed Test

Determine:
The maximum vertical speed of the Pod.

Materials:

- Servo Motor
- Servo Controller Pack
- Winch Assembly
- Rope
- Mass (Weight = Pod assembly weight - Buoyancy force)
- Stopwatch
- Tape Measure
- Foam Padding

Safety:

- Wear Safety Glasses.
- Keep hands away from moving parts when motor is turned on.

Procedure:

1. Place the winch assembly next to the servo motor and make sure that the gear on the motor shaft lines up with the winch chain.

2. Secure the motor in place using bolts to connect the winch frame and the connector plate on the motor.

3. Place the winch assembly on an elevated surface (e.g. Table).

4. Tie the piece of rope to the inside of the winch drum. Make sure the knot is secure. Make 2 - 3 loops of the rope around the winch drum.

5. Tie the other end of the rope to the mass. The weight of the mass should equal the weight of the pod assembly minus the average buoyancy forces exerted by the water on the pod.

6. Measure the distance between the hanging mass and the ground.

7. Cover the ground with foam padding or other soft materials to reduce the impact in case the mass falls to the ground very rapidly.

8. Connect the servo motor the the servo controller pack and power up the controller pack.

9. Step away from the assembly and have the stopwatch ready. Release the brake on the servo.

10. Start the time on the stopwatch and record the amount of time it take to lower the mass to the ground.

11. Repeat the same procedure, but this time run the motor in reverse and record how long it takes to lift the mass back up to the starting position.

12. Take multiple measurements for both directions and calculate the speed using the equation speed = distance/time.

F.4 System Complexity Test

Determine:
To determine that the complexity of the system is low enough to reduce the number of failures due to integration issues.

Materials:

- Pen
- Paper
- Pod

Procedure:

1. Count number of components in Pod (keep track of item name and quantity on paper).
2. Verify the items and quantities are correct by referencing the reports.
3. For any unaccounted items, attempt to relocate them or validate why they are no longer included.
4. For each item on the list, record how and what each item connects to.

F.5 Data Storage Capacity Test

Determine:
Determine that researchers will be capable of storing 10GB - 500GB of data (video, pictures, and locations) in order to conduct research.

Materials:

- Computer

Procedure:

1. Determine the number of researchers allowed to sign up.
2. Determine the number of videos a single researcher will be allowed to record.
3. Determine the number of pictures a single researcher will be allowed to take.
4. Determine the number of locations a single researcher will be allowed to save.
5. Calculate the file size of each video, picture, and location.
6. Look up the amount of storage available on steaming service.
7. Compare the amount of data each researcher uses with the amount of storage available.

F.6 Time to Change Component Test

Determine:
The amount of time to change component. The most time consuming changes will be the components inside of the pod as the pod would have to be opened first, then resealed.

Materials:

- Pod Assembly
- Schrader Valve
- Hand Pump

Safety:

- Wear Safety glasses
- Wear gloves to prevent getting hands sticky with the glue on the pod cap.

Procedure:

1. Secure the pod flat on the ground such that it cannot roll away.
2. Remove the bolted cap from the top of the pod with a wrench.
3. Screw in the Schrader valve in place of the cap.
4. Use the hand pump to slowly pressurize the pod.

5. After every 2 pumps, try pulling the top off the pod. If unsuccessful, pump more air into the pod.
6. The top should slide off when pulled after pressure has reached approximately 5 psi.
7. Record the amount of time it takes to get the pod open.
8. Change a few components inside the pod and record the amount of time it takes to change each component. The components that may require time to change and should be tested are: the LED lights, and the flooding sensor in the bottom of the pod.
9. Reseal the pod by pushing the top in all the way and record the amount of time is takes to do so.
10. Add up the time it takes to open and reseal the pod and the time it takes to change each component to get the total time.
11. Repeat the procedure with different personnel.

F.7 Automated Freshwater Rinse Test

This test procedure is not fully written out since the fresh water rinse system and the corresponding program for it have not yet been developed. This is an outline of what the test would look like and it will be modified later.

Determine:
The functioning of the automated fresh water rinse system. Ensure that the routine would run twice a day..

Materials:

- Fresh water rinse system
- Winch assembly
- Servo motor assembly

Safety:

- Wear safety glasses.
- Keep hands away from moving parts

Procedure:

1. Set up the fresh water rinse system to run at allocated times, that are 12 hours apart.
2. Connect the servo motor to the winch assembly and turn the controller box on.

3. Observe that the servo retracts the winch to the home position at the allocated time

4. The sprinklers should start spraying water after the winch has been retracted and should last for about 5 minutes.

F.8 Video Feedback Delay Test

Determine:
That the video delay caused by network latency is no more than 3 seconds for the pod operator, thus allowing them to see the pod move up and down the pier as they control it.

Materials:

- Two Personal computers

Procedure:

1. Setup host computer to stream video
2. Configure video streaming service to work with video stream
3. Setup client computer to display the stream
4. Use latency monitoring software to determine how long the delay of the video stream is

F.9 Tension Drop Test

The Servo Motor will be have a torque limit set upon it so that it stops when it encounters excessive resistance to movement. The Pulley system implements a hall effect sensor in order to ensure a constant tension on the pod's cable.

Determine:
to ensure that both of these features work as expected in order to prevent damage to the system as a result of extra ordinary load conditions.

Materials:

- Pulley Assembly
- Servo and Winch Assembly
- Rope
- Raspberry Pi
- Weights 5-7lb, 125-130 lb

Procedure:

1. Power On Servo Winch System.
2. Do Not Attach Pod To system
3. Attach a 5-7 lb weight where the Pod Would Normally be Attached
4. Check to see that the hall effect sensor is triggering and the Pul_Out State is entered by the mp2600iec controller.

5. Remove the 5-7 lb weight, and attach the 125-130 lb weight.
6. Attempt to send a move command to the MP2600iec
7. Make sure that the controller does not activate and that the break is not released.

F.10 Camera Pan Range Test

Determine:
To ensure that the pod's Raspberry Pi is capable of communicating with the Pawesome Board to control the pod's rotation for panning the camera and LED's. It will also test to ensure that the hall effect sensor used to prevent over panning works correctly.

Materials:

- Pod Assembly
- Monitor
- Mouse
- Keyboard
- HDMI Cable
- Protractor

Safety:

- Keep hands out of pod when rotating it.

Procedure:

1. Connect the Pod's Raspberry Pi to all the necessary peripherals
   (a) Connect Pi via HDMI cable to a monitor
   (b) Connect USB to mouse and keyboard

(c) Connect Pins to the RS-232 connectors for communicating to and from Pawesome board.

2. Power the Pod, by plugging it into a 120 volt 60Hz wall outlet.

3. Run a test main file that constructs and uses the pod class from the pod.py file located on the raspberry pi in the portal directory. Use the pod's do_function method to communicate panning instructions to the Pawesome board via the Pi's Serial Ports.

4. Test the following instructions:

   (a) rotate left

      i. Ensure pod rotates 5 degrees left each instruction execution

      ii. make sure the pod stops at 60 degree from it's center point

   (b) rotate right

      i. Ensure pod rotates 5 degrees right each instruction execution

      ii. make sure the pod stops at 60 degree from it's center point

   (c) rotate far left

      i. make sure the pod stops at 60 degree from it's center point

   (d) rotate far right

      i. make sure the pod stops at 60 degree from it's center point

   (e) rotate zero

      i. measure the angle from the magnet to the hall effect sensor to make sure it is 60 degrees

F.11 Pod Light Intensity Test

Determine:
To ensure that our lighting system provides the desired intensity, to explore the range of intensities possible with the current system, and determine if the current system can be improved to allow for a better video experience.

Materials:

- Lumen Meter
- Pod Assembly
- Monitor
- Mouse
- Keyboard
- HDMI Cable

Procedure:

1. Connect the Pod's Raspberry Pi to all the necessary peripherals
   (a) Connect Pi via HDMI cable to a monitor
   (b) Connect USB to mouse and keyboard
   (c) Connect Pins to the rs232 connectors for communicating to and from Pawesome board.
2. Power the Pod, by plugging it into a 120 volt 60Hz wall outlet.
3. Run a test main file that constructs and uses the pod class from the pod.py file located on the raspberry pi in the portal directory. Use the pod's do_function method to communicate instructions to the Pawesome board

4. Run the following instructions
   (a) brightness down
   (b) brightness up
5. Stand one meter away from the pod's LEDs and use a Lumen meter to measure the intensity of the LEDs
6. Determine the range of light intensities available with the current lighting system.
7. Does 2000 Lumens lie within the limits?

F.12 System Compatibility Test

Determine:
To verify the developed site behaves reasonably similarly when accessed on different types of PCs and no incompatibilities are encountered

Materials:

- Several PCs connected to the Internet, running Windows, Linux, and Mac operating systems and a selection of the most common browsers, like Firefox, Chrome, Safari, Opera, and Internet Explorer

Procedure:

1. Log on to site with credentials for each of the three different user types on one of the PCs
2. Attempt to perform all actions allowed for the given account type, noting major discrepancies in site appearance, functionality, and performance
3. Repeat steps 1 and 2 for each of the available common web browsers available for that PC's operating system
4. Repeat steps 1 through 3 for each of the PCs

F.13 Concurrent Controlling Users Test

Determine:
To verify there are no conceivable cases where more than one user can have active control of the pod at the same time, which would be an undesirable state.

Materials:

- Several PCs connected to the Internet

Procedure:

1. Log on to site with credentials for each of the three different user types on at least three different PCs
2. Request control of the pod on one of the PCs
3. While the pod is under control of one user, request control on another PC
4. Ensure the second user is given an error message stating someone else is controlling the pod and they must wait to gain control
5. Repeat steps 2 through 4 with every possible combination of user types

F.14 Overhead Test

Determine:
To verify at most 30% of all data sent by the server is overhead, meaning non-video or photo data

Materials:

- 2 PCs
- 3 Cat 5e Ethernet Cables
- Network Hub

Procedure:

1. Connect PC to Hub with one Ethernet Cable
2. Connect Pier Server to Hub with another Ethernet Cable
3. Connect Hub to Pier's network up-link with third Ethernet cable
4. Start Wireshark on PC, filtering for IP Packets
5. Access Pod site from second PC connected either locally or over the Internet
6. Interact with the site for ten minutes, using all available functionality for the given user
7. Stop capture in Wireshark
8. Measure the number of bytes of video and photo data sent by the server, and divide that by the total number of bytes sent by the server
9. Verify the result is at least 0.7

F.15 Video Quality Test

Determine:
Validate that the quality of video being streamed, and saved, over the network is high definition, despite the 1 Mbps limitation the pier presents.

Materials:

- PC
- Raspberry Pi
- Logitech Web Cam
- 2 Cat 5e Ethernet Cables
- Network Switch

Procedure:

1. Connect PC to Switch with one Ethernet Cable
2. Connect Raspberry Pi to Switch with another Ethernet Cable
3. Connect Web Cam to Pi via the USB connection
4. Start Wireshark on PC
5. Start streaming video from Pi to PC
6. Modify the streaming settings such as resolution and color depth
7. Observe effect on Network Usage of the different settings
8. Stop capture in Wireshark
9. Determine network utilization rates for the different combinations of resolution and color depth and determine what is most suitable for the limitation on bandwidth presented by the Pier's Internet connection

F.16 Submersibility Test

Determine:
The ability of the Pod Assembly to maintain it's waterproof characteristic even after repeated sealing and unsealing of the end cap. Previous groups have tested the water-proofness of the pod, though this test is meant to ensure frequent opening of the pod during testing will hinder the waterproof properties of the pod.

Materials:

- Pod Assembly
- Sponge
- Hand pump
- Schrader Valve

Safety:

- Wear safety glasses while removing pod end cap

Procedure:

1. Unseal end cap of pod using Schrader valve and hand pump
2. Remove all Pod internals
3. Place sponge inside pod
4. Re-seal pod

5. Place pod in pool at a depth of 3 meters for 10 minutes
6. Recover pod
7. Inspect sponge, ensure it is dry and that no water leaked into pod
8. Repeat steps 1 through 7 five times, or until a leak is detected

F.17 Accelerated Cable Lowering and Raising Test

There are some concerns with using a plain Ethernet cable rather than a slip ring, specifically, there is the chance that the Ethernet cord will lose its ability to send a clean signal after repeated twisting and untwisting.

Determine:
To determine how long a cable will last being used in such a way, we decided we would automate a test to repeatedly twist and untwist the cable while simultaneously pinging across it.

Materials:

- mpiec 2600iec controller box
- 220V 3phase power Outlet
- SGMGV Servo Winch System
- PC with Python2.7, and Motionworks installed
- Raspberry Pi B
- 2 Ethernet Cables
- Network Switch
- Raspberry Pi portable battery pack

Safety:

- Do Not Approach or try to touch or grab the winch or servo while the servo is powered.

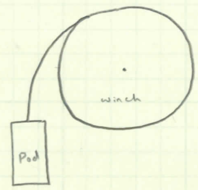
Procedure:

1. Use duct tape to secure the raspberry pi and its battery pack to the winch system.
2. Connect one end of the Ethernet cord to the Raspberry Pi
3. Run the other end of the wire through the winch drum and out of the hole on the side.
4. Connect the end to the switch
5. Connect the PC and the mp2600iec CN1 port to the switch
6. Make sure the mp2600iec EInit switch is on to ensure that it has the expected ip address
7. Power the MP2600iec and the PC and the Raspberry Pi
8. Open Motionworks IEC on the PC
9. Open the Chord_Reliability_Test project file
10. Download the project to the MP2600iec
11. Run Servo_Ethernet_Test.pi
12. Enter the run mode desired
    (a) Continuous
        i. Will rotate the winch clockwise 15 times, then counter clockwise 15 times repeatedly while simultaneously pinging the Pi taped to the winch. The program will stop either when commanded or when the wire begins dropping packets. Upon completion the program will report total number of cycles.
    (b) Count [Value]

i. Will rotate the winch 15 time clock wise then 15 time counter clockwise. It will run through this cycle the number of times entered in place of [Value]. The program will continuously ping the Raspberry Pi. Upon completion the program will report the percent of packets that were successfully sent, and will report the number of failures and their types.

G Analysis

Gear Ratio Verification



Based on previous groups' reports, the amount of force required to lift the pod $\approx 150\ lb_f$.

Normal operating speed of motor = 1500 rpm

$$1500\ rpm \left(\frac{2\pi\ rad}{1\ rev}\right)\left(\frac{1\ min}{60\ s}\right) = 50\pi\ rad/s$$

Motor Torque @ 1500 rpm $\approx$ 3 N·m (from data sheet)

Winch Torque = Motor Torque * gear ratio.

$$\text{Winch Torque} = (3\ N{\cdot}m)\left(\frac{20}{1}\right)\left(\frac{7}{1}\right) = 420\ N{\cdot}m$$

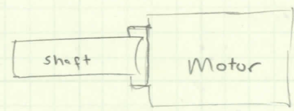
$$\text{Force} = \frac{T}{r} \approx \frac{400\ N{\cdot}m}{10\ in} \approx 40\ \frac{N{\cdot}m}{in}$$

$$40\ \frac{N{\cdot}m}{in}\left(\frac{1\ ft{-}lb_f}{1.356\ N{\cdot}m}\right)\left(\frac{12\ in}{1\ ft}\right) \cong \underline{354\ lb_f}$$

More than enough

Critical Speed

start with crude approximations; refine if necessary



Critical speed; $\omega = \left(\frac{\pi}{l}\right)^2 \sqrt{\frac{g E I}{A \gamma}}$

Assume steel shaft

$E = 30 \text{ Mpsi}$

$\gamma = 0.282 \text{ lbf/in}^3$

$A = \frac{\pi}{4}(D)^2 = \frac{\pi}{4}(1.25 \text{ in})^2$

$I = \frac{\pi}{64}(D)^4 = \frac{\pi}{64}(1.25)^4$

$l = 3 \text{ in.}$

$$\omega = \left(\frac{\pi}{3 \text{ in}}\right)^2 \sqrt{\frac{32.174 \frac{\text{ft}}{\text{s}^2} \cdot 30\times10^6 \frac{\text{lbf}}{\text{in}^2} \cdot \frac{\pi}{64}(1.25)^4 \text{ in}^4 \left(\frac{12 \text{ in}}{\text{ft}}\right)}{\frac{\pi}{4}(1.25)^2 \text{ in}^2 \cdot 0.282 \frac{\text{lbf}}{\text{in}^3}}} \left(\frac{1 \text{ rev}}{2 \text{ rad}}\right)\left(\frac{60 \text{ s}}{1 \text{ min}}\right)$$

$\omega \cong 663{,}000 \text{ rpm}$

well away from normal operating conditions. Load carrying bearings along shaft not needed.