360 Webcams for Zoos and Aquariums

Design Document

Version: 2.0.0 – 2018.04.24

General Information

Team Number: sddec18-12

Client: *True 360* (Christopher James → chris906@iastate.edu)

Faculty Advisor: Dr. Henry Duwe (duwe@iastate.edu)

Team Members and Roles:

Name	Primary Role	Secondary Role(s)
Nathan Cool	Front-End Engineer	Project Manager, Webmaster
Zach Newton	Front-End Engineer	Scrum Master, QA
Ian Jamieson	Back-End Engineer	Graphics Lead
Alan Negrete	Back-End/Database Engineer	Scribe, QA
Tarek (TJ) Yacoub	Embedded Engineer	Communication Lead, QA
Hosam (Sam) Abdeltawab	Embedded Engineer	Software Architect

Team Email: sddec18-12@iastate.edu

Team Website: sddec18-12.sd.ece.iastate.edu

Table of Contents

List of Figures

1 - Introduction

1.01 - Acknowledgement

We would like to acknowledge our client, Chris James, and our faculty advisor, Dr. Henry Duwe, for their contributions to our project. Chris has allocated approximately $20,000 and contributed multiple 360° webcams and other hardware components for our team to use during development. He has also dedicated significant time to meet with us every week as we develop the project. Dr. Duwe has contributed his time and technical advice to our team, as well as assisted in steering us in the right direction on multiple occasions.

1.02 - Problem Statement

360 Webcams for Zoos and Aquariums is the brainchild project of Chris James, founder of the Ames, Iowa-based startup company, True 360. The purpose of the project is to provide zoos and aquariums with an easy-to-use 360° webcam system for educational, animal health, and business/marketing purposes.

As of now, there are three problems which True 360 hopes to solve with its 360° webcam project. The first problem is that zoos and aquariums do not have access to an easy-to-use 360° webcam solution for their exhibits. While there *are* zoos and aquariums that currently use webcams (even 360° webcams) for various purposes, no solution currently exists which would allow them to connect and control all of their webcams from a central (remote) location. The second problem is that zoos and aquariums are always looking for ways to boost their web and social media marketing techniques to both attract new visitors and improve public awareness of issues such as animal health and wildlife conservation. The third and final problem is that zookeepers, animal health professionals, and other staff members are currently required to visit each exhibit on a regular basis in order to monitor the animals. This process is time consuming, which means staff members have less time to focus on other responsibilities.

With the 360 Webcams for Zoos and Aquariums project, our team—in collaboration with True 360—aims to provide solutions to these problems. By developing a system which will allow zoo and aquarium staff members to connect and remotely control multiple 360° webcams, the necessity of physically interacting with each webcam will be removed. The system will allow for webcams to be installed in both above-ground (indoor and outdoor) and underwater exhibits and controlled via a central web application. Conveniently, this application will help us solve the aforementioned marketing/awareness and animal health problems. Our system will enable zoos and aquariums to constantly capture 360° footage of animals in their exhibits, which can then be used for educational live streams, creating promotional content for their websites and/or social media accounts, monitoring animal health and behavior, and archival purposes.

1.03 - Operating Environment

The major physical components of the system include one or more 360° webcams and any central computing hardware necessary for connecting/controlling the webcams and managing the captured footage. In order for the system to maintain a normal operational state, there are many environmental factors which must be taken into account.

Zoos and aquariums will place webcams in a variety of locations, including both above-ground (indoor *and* outdoor) and underwater exhibits. Webcams placed above-ground and outdoors must be able to operate while exposed to the local weather. Webcams placed above-ground and indoors must be able to operate while in glass-covered exhibits or in areas where visitors' devices may tax the network (if public/staff network traffic is not separated). Exposing electronics to underwater conditions inevitably poses many potential problems which will have to be addressed (e.g., cable exposure, underwater webcam temperature regulation).

Regardless of where webcams are placed, there are many universal environmental factors which must be considered in order to maintain a normal operational state. The webcams must have constant access to ample power, a stable and strong (wired or wireless) Internet connection, and a case to physically protect the webcam unit. Fortunately, our client will be responsible for the process of fabricating a case to fit the webcam model used in the system.

One of the most important environmental factors which must be considered is the presence of animals. The webcam unit must be durable enough to withstand the impact and pressure of strikes, bumps, bites, and other animal interactions. Any cables (e.g., power, Ethernet, USB) connected to the webcams must be installed in such a way as to remain out of the sight/reach of the animals.

In addition to the webcams, we must take into consideration the computing/storage hardware and the software (web application) components and their respective environments. Any computers (desktops, servers) used must be stored in a secure indoor and temperature-regulated environment in order to avoid issues such as overheating and unauthorized access. The web application must be developed with digital security in mind.

Overall, these environmental factors represent our basic understanding of and assumptions about how zoos and aquariums will use the system. They do not encapsulate every possible factor, as there will inevitably be some variance between facilities/users.

1.04 - Intended Users and Uses

Three categories of end-users must be taken into account when developing the system:

Zoo Marketing Directors
The role of a zoo marketing director is to promote the zoo through various forms of media. It's absolutely essential that we make this product easy enough to use such that the marketing director can utilize the software to its fullest potential. The zoo marketing directors will be able to embed or publish (to YouTube) 360° live streams of animal exhibits. The marketing director is also interested in publishing the content created by our product through various social media channels.

The marketing director and other zoo staff will be able to receive relevant video and photos through the marketing platform, as well as through a weekly email. The marketing director will likely utilize the photos and video to showcase the zoo in a rarely seen before way.

Zoo Curators
The primary role of zoo curators is to monitor and care for the animals in their exhibits. The zoo curators will be able to monitor animals without the need to go to the desired exhibit. Ideally, this will improve the zoo curators' effectiveness because they will be able to be in more places than before. Additionally, curators will be able to go back in time—since streams will be digitally archived—to learn more about animals' everyday patterns.

As mentioned above, the role of a zoo curator is to monitor and care for their animals. Another important feature of our product is the activity monitoring capabilities. Our product will be able to alert zoo curators in near real time if an animal has been abnormally active or less active than usual.

Zoo IT Departments
One of the main issues that has come up throughout our initial research is that zoo IT departments are relatively small. This brings up the question of how will the IT departments be able to keep up with the new demand of the 360 webcams in their zoo. Our solution to this is to provide an easy to use and managed administrative portal where the zoo IT staff can remotely control the cameras.

1.05 - Assumptions and Limitations

Assumptions
Our system *will not* be responsible for:

- Financial transactions (facilitation, record-keeping)
- Communication-related functionalities

Our system *will* be responsible for:

- Webcam management
- Video archiving (via a connected cloud-based service)
- User management (multiple user types)

Technical Assumptions:

- Webcams will be powered via standard 120V wall outlets
- Internet in zoo is fast enough to stream live video
- Zoo has IT staff to set up the cameras

Limitations
Our system will assume the following limitations:

- Reliance on manufacturer-provided APIs and SDKs to control the webcams, set up live streams, etc.
- Limited connection mediums to webcams via WiFi, Ethernet, or USB
- Potential bottlenecks when collecting high-resolution footage (1080p, 4K) due to server storage cost/space and Internet bandwidth at zoos and aquariums

1.06 - Expected End Product and Other Deliverables

360° Live Stream Webcam System

Estimated Delivery Date:

- May 2018 (Version 1), December 2018 (Version 2)

Description:

- Our team will select hardware/software components appropriate for developing 360° webcam software. These components will be selected from amongst existing industry products (webcams, servers, protocols, database frameworks, front-end frameworks, etc.).

Administrative Portal Web Application

Estimated Delivery Date:

- May 2018 (Version 1), December 2018 (Version 2)

Description:

- Our team will develop a web application which will allow zoo and aquarium staff members to connect to and remotely control the webcams installed in their exhibits. The web application will be optimized so as to provide accessibility on both desktop and mobile web browsers.

Animal Health Monitoring Web Application

Estimated Delivery Date:

- December 2018 (Version 1)

Description:

- Our team will develop a web application which will allow zoo and aquarium staff members to monitor the health and well-being of animals. It will utilize machine learning to observe normal animal behavior and alert staff members of abnormalities.

Marketing Web Application

Estimated Delivery Date:

- December 2018 (Version 1)

Description:

- Our team will develop a web application which will allow sponsors to integrate company logos and product advertisements into live streams. The web application will also allow zoo and aquarium marketing staff to extract images and video clips for marketing purposes.

2 - Specifications and Analysis

2.01 - Proposed Design

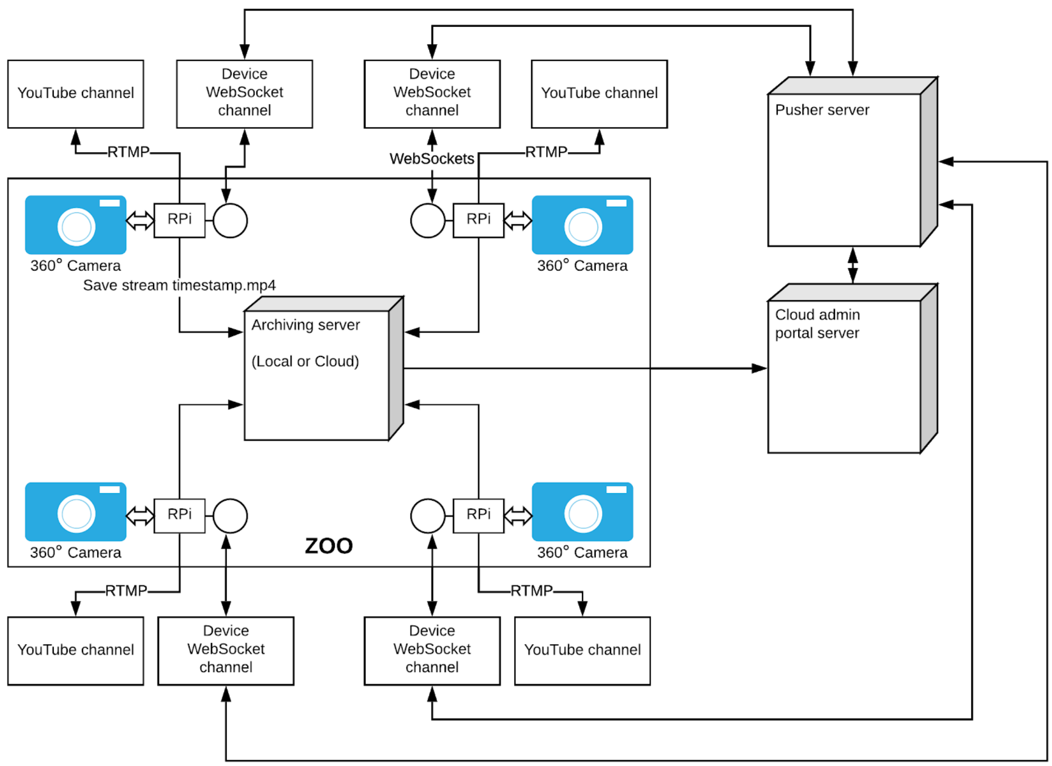

Figure 1: Proposed Design

How It Works
Each 360° camera is behind an interface provided by the Raspberry Pi. Each Raspberry Pi will have its own websocket channel that it is subscribed to, where the backend can trigger events on, which in turn the Raspberry Pi will send the command related to that event to the 360° camera. The Raspberry Pi will run the FFmpeg library (transcoding software) which streams videos to youtube as well as archive to a given path.

360° Webcams
The Garmin Virb is compacted live streaming webcam that provides good accessibility for developer to develop software around it. The webcam provides a REST api through WIFI. The Garmin Virb is able to stream and take still pictures in different resolutions that the developer can change through the provided REST api commands.

Storage Solution
Local or Cloud based solution that will archive 72 hours of previous live stream from each webcam. Consequently, zoo staff will be able to go back at most 72 hours to address any issues related to animal exhibits. Finally once a live stream is complete the machine learning algorithm will be executed here.

Admin Web Application
Since different zoos will be using the same admin portal, the web application will be hosted on the cloud. Users will be associated with a zoo, therefore, when a user logs in the Admin Portal they will be able to manage (depending on the user permission) the system related to the zoo he/she works in. Finally, users will be able to request previous live stream if needed.

Other
The Garmin Virb 360 is able to stream in 4K resolution. Additionally, the Raspberry Pi handles synchronous streaming and archiving by using the FFmpeg library. The web application will use Firebase Authentication to prevent people from messing around with zoo's stream.

2.02 - Design Specifications

Functional Requirements

360° Webcams
- Stream live video in various resolutions.
- Capture photos in various resolutions.

Storage Solution
- The system must be able to store past live streams for archival purposes.

Admin Web Application
- A user system must be built-in to control access to various functionalities.
- IT staff must be able to control/monitor the webcams.
    - Turn webcams on and off
    - Adjust webcam settings
    - Start/stop live streams
    - Insert delays between capture and live stream
    - Capture photos
    - View webcam health/status properties
- IT staff must be able to view archived live streams.

Animal Health Web Application
- Monitor animal activity remotely
    - See animals activity over a selected period of time
    - Utilize machine learning models to detect animal movement
    - Revisit times where the animal was particularly active if an issue arises
    - Send alerts to zoo curators when an animals activity level has abnormalities
- This feature will not be properly addressed until fall 2018.

Marketing Web Application
- Marketing staff must be able to integrate logos and other advertising materials into live streams through the web application.
- Sponsors can "rent" allotted times for selected streams to display their logos and other marketing/advertising materials.

- This feature will be integrated into the web application during fall 2018.

Non-Functional Requirements
- Webcams must be able to operate in various different environments.
- The system must use a service such as AWS snapshots to backup back-end functionality.
- The back-end should rely on AWS to properly scale as the service grows.
- The system should be able to be remotely maintained via Snappy for Ubuntu Core.

2.03 - Design Analysis

Front End
- Details: Our system's front-end component consists of a web application. This web app allows zoo staff members to interact with their cameras, stream archives, etc. Ideally, the web app will be easily accessible via any Internet-connected device, whether it be a computer, a tablet, or a phone.
- Status: The front-end is still in development. A vast majority of the work is close to being completed. We're getting close to the point of
- Going Forward: We are continuing to build the framework for the True 360 website and identify the minimal viable product.
- Dependencies: We are currently using React and Redux.

Back End

Node JS Server
- Details: We have developed our first prototype on the back end using Node JS. We have competed setting up the project, and the API is fully functional. The project has been broken down into a MVC structure. We have also implemented an ORM design to ensure modularity of the project going forward and to allow us to expand on our API functionality quickly and efficiently. Currently we have set up the migrations, models, seeders and controllers based on our Database Diagram below.
- Status: So far, it appears that this approach will work well. We will continue expanding on this prototype and analyze it to make sure we start doing a solid best practices approach design, instead of focusing on just making it work.
- Going Forward: We will begin expanding on the API endpoints, allowing our React front end to easily modify & request data for Zoos, Users, Exhibits and Cameras. Another next piece we want to focus on is the authentication aspect, we will use Firebase Authentication on the front end and add a middleware on the back end using the Firebase Admin package to ensure our API is secure from unauthenticated users. Lastly, we will start generating documentation for our API. This will include details for each controller, such as the routes, request types, request parameters and some expected response examples.
- Dependencies: Our primary depency is Sequelize, which is what we use for our ORM design. We'll have to dig deeper and test this library, to ensure it will fulfill our future requirements as well. If it can't, we would then be able to quickly look for an alternate solution, while it's still early in the project. We are also using

Faker for seeding our database during development. Along with the other popular libraries that work with Node, such as express.

Database

- Details: Our initial prototype is using MySQL for the database. Currently, our design only includes the core aspects of the application. We begin expanding as the project progresses. This is something that we plan to do going forward. For now, we want to get the core functionality of the application and ensure there is a good communication between our software and hardware components. From there, we will expand on the React side of things, along with the necessary additions to the back end and database.

- Diagram:

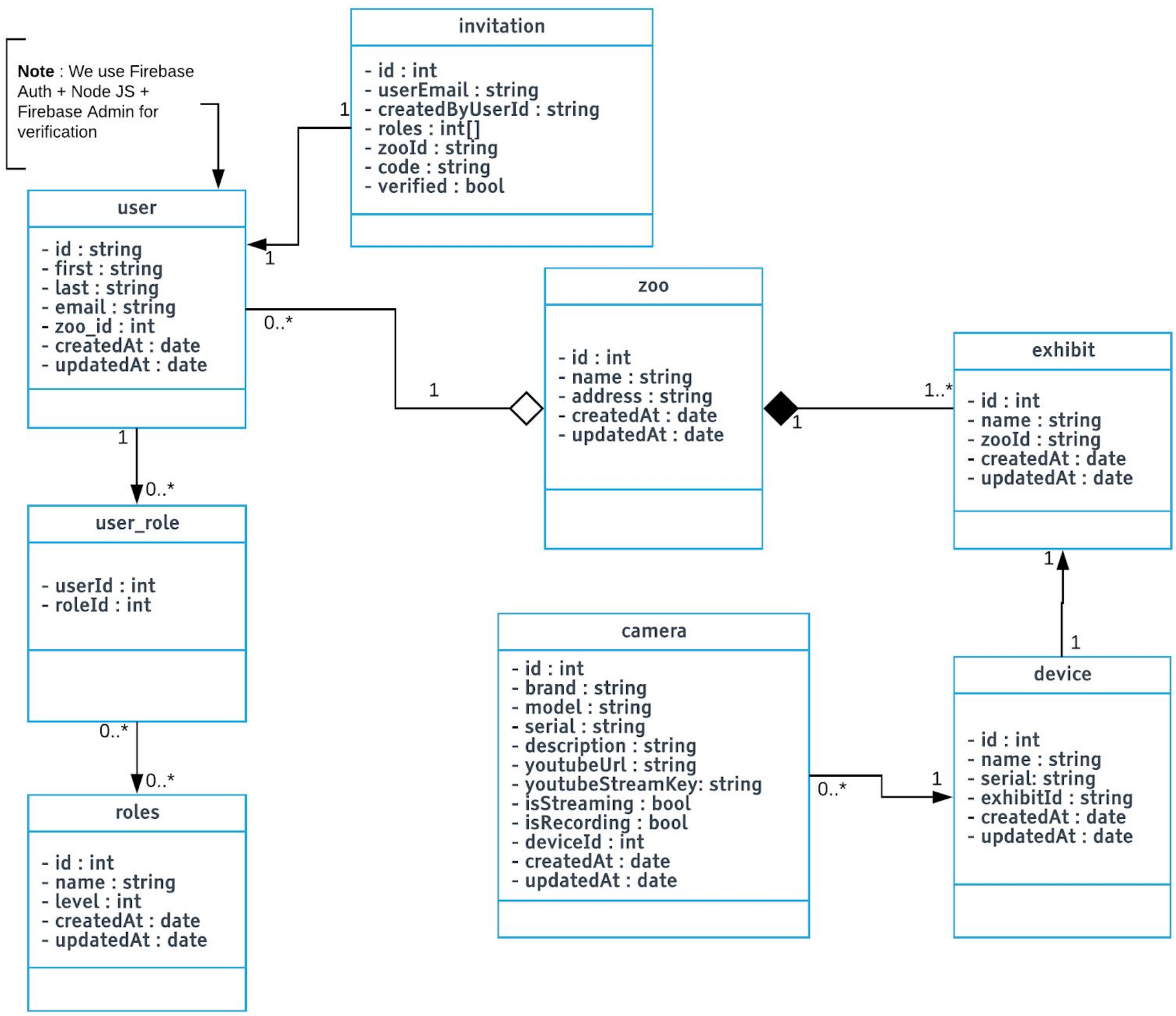

Figure 2: Database Design

Embedded

- Details: As of the time of writing this document, the main functionality of the embedded portion of this project is complete.
- Status: The main functionalities implemented are registration, start stream, and stop stream.
- Going Forward: There are more functionalities that will be implemented but not necessary for the prototype such as temperature sensor readings, camera locator, etc. Moreover we will be using snappy, a package management system for Ubuntu Core to ensure remote maintainability.
- Dependencies: The application on the FFmpeg library and Pusher client package.

3 - Testing and Implementation

3.01 - Interface Specifications

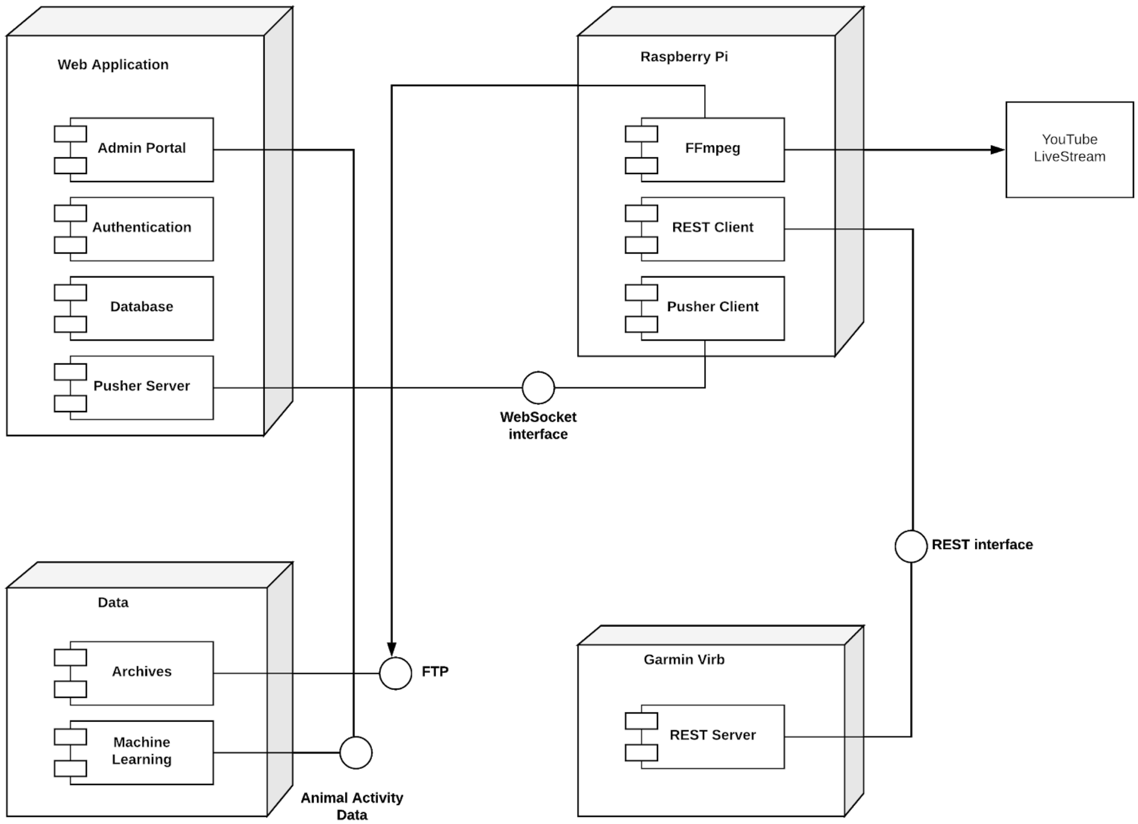

Figure 3: Interface Design

Firebase Authentication
Our web application will need to communicate with an authentication service in order to provide security for user login purposes. We are using the Firebase Authentication service, which will provide our web application with a straightforward JavaScript API interface through which to store credentials.

WebSockets
We are using WebSockets communications to interface between the Admin Portal and the Raspberry Pi.

REST API
The REST API will be used to interface between the Admin Portal and the zoo server.

FTP
We will be using File Transfer Protocol to save live streams to our archiving solution.

3.02 - Hardware and Software

Hardware

- Garmin Virb 360 Camera: The Garmin Virb 360 is a commercially available 360 camera. It is being used for minimal viable product and test any new ideas we think of along the way.
- Raspberry Pi 3: The Raspberry Pi is being used to store and stream the video from the camera. The Raspberry Pi will also power the camera via a USB cable. The Raspberry Pi will then be connected to an ethernet cord to power both the Raspberry Pi and the camera controller.
- Zoo archive (local): The zoo archive will hold 72 hours worth of archived video. The zoo archive will also house the image processing and activity monitoring aspect of the project.

Software

- NodeJS: We used, and are still using, NodeJS for the server side technology.
- ReactJS: A Javascript framework for frontend development.
- FFmpeg: Used to transcode from RTSP stream to RTMP stream.
- Pusher: WebSocket service that we use to interface between the Admin Portal and the camera controller.
- Cameras' SDK & APIs: Many of the cameras that we've worked with have SDKs or APIs available for development purposes. We've been using the ones available to us for the testing phase of the development process.
- MySQL Database: We will be using a Relational Database to for storing all persistent information regarding the application. In addition, with Sequelize, we have migrations that we can quickly run on any Server hosting our Database.

3.03 - Process

General

We plan on using the GitLab continuous integration and continuous development tools for testing builds of the entire project. Any time changes are committed to a branch, our testing server or virtual machine will start a build. Any pull requests will also be required to have a successful build before being merged into the master branch.

Front-End

The front-end testing process will involve the use of a build server. Whenever code changes are pushed, the server will allow us to build the project and run various types of tests to validate the new features. We will focus on both code-based and interaction-based tests.

Back End

Our focus at the moment is to ensure that we have a solid foundation set up so that the other teams are able to continue development. Next semester we will focus on implementing an API testing framework to run tests during merges, thus ensuring we don't break any existing API Routes as we develop and that additional routes are also tested on the fly.

Embedded
Our hardware programming will be tested in a few different ways, but at the moment, we are just doing verification and validation testing. This has allowed us to properly ensure that the product that we are building is meeting expectations for the project.

Flow Diagram

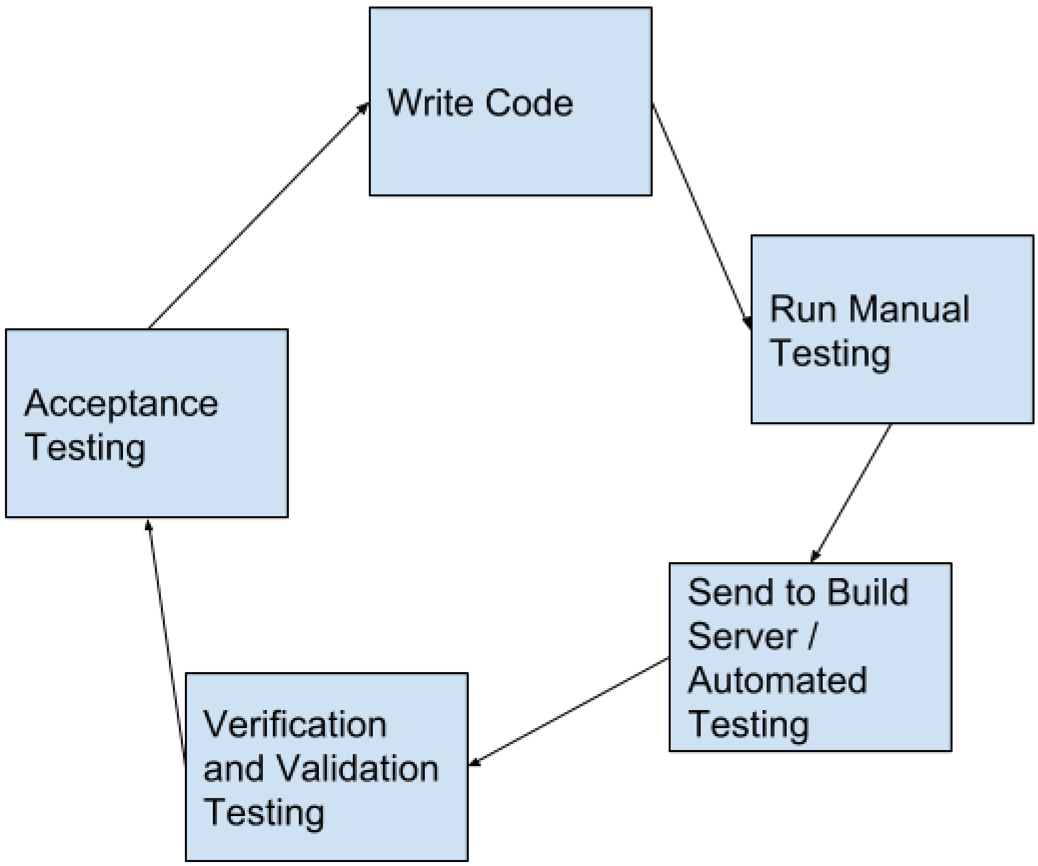

Figure 4: Process Flow Diagram

System Testing

<u>Archiving Video Test</u>

- Goal : Meet functional requirement, “ IT staff must be able to view archived live streams”
- Verification Process
    1. Live stream and archive video for 8 hours
    2. Ensure video all video files have been stored in server at zoo
    3. Check if all videos are accessible from the Web App
    4. Check if a specified date & time is accessible

<u>Outage Test</u>

- Goal : Meet functional requirement, “IT staff should not have to physically access the devices after outages to get stream back up ”
- Verification Process
    1. Disconnect & reconnect power / network from device or camera
    2. Verify that device restarts and is listening to correct WebSocket Channel

3. Send a trigger/command from the Web App to the device
4. Ensure that device picks up command and executes it

API Server Crash

- Goal : Meet non-functional requirement, “Application Services must be reliable ”
- Verification Process
    1. Force restart server or kill the program
    2. NodeJS server should automatically restart
    3. Connection to Database & Pusher should be available via configurations
    4. Hit API Endpoints via Web App or Postman to verify service is back up

3.04 - Results

Throughout the duration of the semester so far, we’ve had a few key opportunities to test what we’ve been working on and the ideas we’ve come up with. As discussed above, we tried quite a few different types of cameras and technologies. This has allowed us to get a better understanding of the market, what’s available, and what we can develop to fill the need.

In testing our camera options, we’ve come to the conclusion that the Garmin Virb 360 is the best camera for the time being. This has allowed us to begin development with this camera and make progress towards a minimal viable product. That being said, we’re still working towards better testing methods which will allow us verify the quality of the product we’re producing. The next few months will be focused on building the product out and developing testing environments for the technology.

4 - Closing Materials

4.01 - Conclusion

In collaboration with True 360, our team aims to provide zoos and aquariums with an easy-to-use 360° webcam system for educational, animal health, and business/marketing purposes. Each of these purposes will provide new kinds of business value to zoos and aquariums.

Webcam setups currently used by zoos and aquariums provide limited functionality with respect to interconnectivity, footage quality, and software features. Many of the solutions that are currently on the market do not have the same 360° footage that we will be providing.

Our solution will remove the need for physical interaction with each webcam and provide a wide array of software features including centralized webcam control, 360° footage capture/archiving and live streaming, animal health monitoring, and curation of marketing content. Our team will be developing this functionality over the course of two semesters.

While the underlying technology has been around for years, we hope to utilize new tools to build a new 360° webcam system for zoos and aquariums. They will be able to interact with their exhibits like never before, hopefully being able to utilize these new technologies in effective ways.

4.02 - References

"Build Something with Us." *Home | Garmin Developers*, developer.garmin.com/.

Garmin, and Garmin Ltd. "Action Cameras | VIRB 360." *Garmin*, buy.garmin.com/en-US/US/p/562010.

"Home." *Open Broadcaster Software*, obsproject.com/.

"Product." *RICOH THETA*, theta360.com/en/about/theta/v.html.

RICOH THETA Developers, developers.theta360.com/en/.

"Transport Your Audience:VR in 8K." *Insta360 Pro - Transport Your Audience: VR in 8K*, www.insta360.com/product/insta360-pro.

"True 360." *True 360*, true360.weebly.com/.

4.03 - Appendices

N/A – Stay tuned.

4.04 - Changelog

- **2018.04.24 (Version 2.0.0)** – Second draft submitted
- **2018.03.05 (Version 1.0.0)** – First draft submitted