360 Webcams for Zoos and Aquariums

Project Plan

Version: 3.0.0 – 2018.04.24

General Information

Team Number: sddec18-12

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

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

Team Email: sddec18-12@iastate.edu

Team Website: sddec18-12.sd.ece.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

Table of Contents

List of Figures

List of Tables

1 - Introduction

1.01 - Acknowledgements

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 - Purpose, Problems, and Solutions

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 - Proposed Approach and Statement of Work

2.01 - 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.

2.02 - Constraints Considerations

Constraints

- Our product must be able to function in environments with varying temperatures.
- Many of the webcam options are not fully developed or do not supply APIs/SDKs for development purposes.
- The system requires significant bandwidth to ensure stable and high-quality live streams.
- Most webcam options require the use of OBS as a middleman, which limits the amount of custom functionality we can implement.

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 monitored via Snappy for Ubuntu Core.

2.03 - Technology Considerations

360° Webcams

- Insta360 Pro – The Insta360 Pro is a professional-grade webcam which provides a wide variety of I/O that we could use to stream videos. Moreover, the quality of the stream is superb and the webcam is able to stream continuously for longer intervals than the other two cameras. Unfortunately, the company that makes the Insta360 Pro does not provide an API or an SDK for developers, restricting us to use of the webcam's proprietary application.
- Garmin VIRB 360 – The VIRB is significantly cheaper than the Insta360 Pro; therefore, it is possible to place more webcams in an exhibit at a lower cost. Garmin provides a complete API for the VIRB which works as advertised. Unfortunately, the camera does not have an Ethernet input (for faster streaming) and only has two lenses, which results in noticeable stitching on the live stream. Moreover, it proved to be difficult to publish the live stream to YouTube since the streaming protocols are incompatible.
- Ricoh Theta V – Like the Garmin VIRB, the Theta V is significantly cheaper than the Insta360 Pro. The Theta V provides more control of the camera compared to the VIRB, as it has a REST endpoint as well as MTP protocol that is used to control the camera through USB. Moreover, the camera has two lenses, which results in a noticeable stitching effect on live streams.

Front-End

- Open Source Frameworks
  - Pros – Access to pre-built UI components, supported long term
  - Cons – Not as customizable
- React
  - Pros - High performance, inherently modular, extensible
  - Cons - Large boilerplate to start, medium learning curve
- Redux
  - Pros - Single-source of truth, easier debugging
  - Cons - Large amount of boilerplate, large learning curve
- Custom Theming Frameworks
  - Pros – Customizable to our liking
  - Cons – More work to build from scratch; we would need to keep up with web conventions to ensure our front-end application adheres to expectations/standards.

Backend

- NodeJS
  - Works on a non-blocking I/O model which makes it easy and clean to use.
  - Scalable
  - Can be used to access the new NoSQL technologies such as MongoDB and CouchDB.
  - Has predefined modules for server side development which makes it easier to write function calls on the server.

Server

- Cloud Solution
  - Scalable
  - Medium latency
  - Less expensive in the long run (due to elasticity)
- Local Solution
  - Low latency
  - No bandwidth cost
  - More control of zoo content
- NoSQL
  - DynamoDB
  - MongoDB
  - CouchDB
  - Firebase Realtime DB
- Relational DB
  - It's very difficult to create full NoSQL applications. They are often a hybrid, taking advantage of both technologies. Many complex queries are impossible without relations.
    - AWS Relational DB
    - MySQL server installed on EC2

Storage

- Local Storage
  - RAID SSD 4TB Drive
  - SSD Hard Drive
  - RAID 4-6TB HDD Drives
    - Similar transfer speeds to SSD in RAID 0
    - Main difference is access time, but transfer speeds are the priority for large data.
- Cloud Storage
  - AWS
  - Google Cloud Services

2.04 - Safety Considerations

Operating Environment

- Monitoring the system to ensure safe operating temperature and avoid overheating.
- Make sure all wires are installed where animals can not interfere with them.

System Security

- Make sure the application is secure and will not be vulnerable from outside attacks.

Time Buffer for Live Video

- Make a time delay for the live video, so if something were to happen to an animal, the zoo staff can prevent that from being streamed.

2.05 - Previous Work and Literature

There are currently some zoos (see references) that provide live-streaming webcams on their websites. For example, the San Diego Zoo has live streams for some of their exhibits. The main difference is that many of them are low quality webcams and they are not 360°. Moreover, the streams are unstable and often time-out or disconnect. It's worth noting that they also did not utilize the YouTube platform for their streams. Finally, there appears to be no sound coming from the streams, which could be important when analyzing the animal behavior.

Advantages of the San Diego Zoo Architecture

1. Zoo staff won't need to maintain multiple YouTube channels to publish more than one live stream.

Disadvantages of the San Diego Zoo Architecture

1. The viewers of the stream are unable to go back to a specific moment in time and watch things that happened on the zoos Youtube channel.
2. Viewers are unable to interact with other viewers using comments, which could assist in attracting more viewers in the long run.

2.06 - Possible Risks and Risk Management

Potential Risk – A team member is considering dropping the course.
Management – Maintain open communication amongst team members in order to address and resolve any issues/impediments in a supportive manner. In the event that the situation occurs, communicate with course instructors, our faculty advisor, and the team member in order to mitigate any potential impacts on progress.

Potential Risk – The client loses interest in the project or the project is discontinued.
Management – Team members should think about contingency plans and/or alternative projects. In the event that the situation occurs, immediately contact course instructors and our faculty advisor to determine how to proceed.

Potential Risk – A major hardware component is damaged.
Management – Team members should take care to ensure proper storage, transportation, and usage of all hardware component. In the event that the situation occurs, document it immediately and notify the owner of the damaged component in order to determine whether it can be fixed/replaced under warranty or if complete replacement is feasible.

Potential Risk – Members of the team deviate from the project schedule or team productivity decreases.
Management – Ensure that everyone is on task at all times by utilizing our communication and planning tools and procedures (see subsection 2.08). In the event that the situation occurs, call a meeting to discuss the deviation and identify the necessary steps to return to the proper project schedule (and mitigate any potential losses).

Potential Risk – Research efforts are consuming significant time.
Management – Research is inevitably part of the learning process; however, if research efforts are not leading to the desired results, consider reaching out to other members of the team, pulling in an external expert/consultant (a service offered by our client), or reaching out to our faculty advisor or another faculty member for support.

Potential Risk – Sudden changes in requirements (scope creep)
Management – In order to prevent scope creep, the best course of action is to define a concrete set of requirements and obtain written agreement (signatures) from all parties. In doing so, potential instances of scope creep can be mitigated.

2.07 - Project Proposed Milestones and Evaluation Criteria

Our project will contain multiple milestones, most of which will consist of successfully implementing major hardware/software functionalities. Below is the current list of proposed milestones, based on our progress thus far:

- **Milestone 0 (Pre-Development)** – Successfully identify compatible hardware and software components which will allow us to develop the 360° webcam system.
- **Milestone 1** – Successfully develop software which provides a basic interface between a 360° webcam and a live (YouTube) stream.
- **Milestone 2** – Successfully develop software which allows users to remotely control a webcam.
- **Milestone 3** – Successfully develop software which allows for video storage (archiving).
- **Milestone 4** – Successfully develop software which allows logos and advertisements to be embedded in a live stream.
- **Milestone 5** – Successfully develop software which uses machine learning to observe animal health and behavior.
- **Milestone 6** – Successfully develop software which assists in curating potential marketing materials from existing video archives.
- **Milestone 7 (Post-Development)** – Successfully deploy the system at a zoo/aquarium.

For details regarding evaluation criteria (test cases), see Appendix B, Table 1.

2.08 - Project Tracking Tools and Procedures

Our team will utilize the following software tools:

- GitLab + Gitlab Issues – Code repository and task tracking system
  - GitLab Boards Broken Up into Two-Week Sprints
    - Each issue is tagged with the appropriate Sprint it will be worked on, the date that it's due.
    - All changes to an issue will be reflected with a new tag as appropriate.
    - Boards are aligned with our Project Timeline. This makes simple to do Project Management and will allow us to quickly notice if we're falling behind.
- Google Drive – Document storage
- Lucidchart – Diagram storage
- Slack – Group communication

Our team will utilize the following procedures:

- A hybrid "Agile" development process:
  - Each group member posts on our *#daily-standups* Slack channel
    - "What did I do yesterday?"
    - "What will I do today?"
    - "What impediments are in my way?"
  - Two-week sprints with four official meetings

        - The first Monday of a sprint = planning day
        - The first Thursday of a sprint = progress updates
        - The second Monday of a sprint = progress updates
        - The second Thursday of a sprint = present progress and finalize sprint
    - Dedicated scrum master to manage Gitlab tasks
    - Emphasis on quick turnaround for task completion to facilitate client feedback

- <u>Status Reports:</u>
    - Tracks weekly progress (past, present, and future) for each team member
    - Tracks weekly hours contributed by each team member
    - Tracks weekly issues/impediments which may arise
    - Uses a consistent, easy-to-follow template
    - Posted on our team's website every week

- <u>Coding Standards:</u>
    - Create modular code
    - Comment the code that is written
    - Push to Git often
    - Reduction of time complexity
    - Frequently refactoring design and understanding of design

2.09 - Objective of the Task

By the end of project, our team must have the following objectives complete:

- A web application that is capable of sending commands to a desired webcam, these commands include but are not limited to:
    - Starting a stream
    - Stopping a stream
    - Take still pictures
    - Change stream quality, and other related video/image settings.
- The web application should provide the capability of integrating the live stream to youtube.
- A system that transcodes streams from the webcam to a format digestible by youtube and web pages in general.
- Provide a per zoo service to store streams for a desired amount of time to give zoos the ability to go back in time.
- A process that will embed sponsors logos in the live stream.

Other objectives that our team has in mind, and will be implemented depending on the time frame:

- A machine learning service that can identify some animal concerns such as activity.
- Automatically provide marketing assets to the marketing team at each zoo.

2.10 - Task Approach

Zoos ideally will have multiple webcams, these webcams will be connected to the same network. These webcams are controllable through the web application, this is possible because most webcams have a REST API.

Our first design approach is a localized architecture, hence having a local transcoding server and a storage solution. See Appendix A, Figure 1 for a graphical representation of the localized design approach.

Our second design approach is a cloud based solution, where we could utilize services such as Amazon aws, and google cloud platforms to deploy our transcoding server and cloud storage. See Appendix A, Figure 2 for a graphical representation of the cloud design approach.

The functionality is similar in both designs. The purpose of the transcoding server is to digest the video output of the webcams and convert them to a format we could use on youtube, and web pages. Having the transcoding server locally would result in lower latency and no bandwidth costs that are resulted from uploading the videos to a cloud platform. On the other hand the cloud solution is scalable and more future proof. The same applies to the storage solution but is not expected to perform as high of a standard and the transcoding server because it is used to go back through streams when need be.

This is the current and final design approach. Instead of a transcoding server, now we have a camera controller that will handles the transcoding of the footage from the camera, as well as handling the commands that will be received from the Admin Portal. We decided to take this approach because zoo owners had expressed that they do not like the idea of having a server to maintain. Therefore, we decided to go with the Internet of Things approach because it is a lot cheaper than having a server, and we could easily push update to our devices in zoos. See Appendix A, Figure 3 for a graphical representation of the controller design approach.

2.11 - Expected Results and Validation

Our expected results and validation are closely related to our project milestones. For each of the milestones we expects the following:

- Camera Controller - The controller should accommodate various internet speeds by changing the streaming quality of the webcams. Also the controller should be able to handle different streaming protocols in case we have changes in the future. Moreover, the controller should be able to receive commands from the admin portal and execute the desired functionality in real-time and with out errors.
- Zoo Admin Portal - Webcam should be controllable through the web application with low latency. The connectivity medium of the webcam should not matter.
- Video storage- Be able to archives videos for a desired amount of time. Provide an intuitive way to retrieve videos for a specific timeframe.

- Logo Integration- Logos should be displayed in the right place with not warping effects.
- Marketing App - Provide a place where sponsors can place orders to have the advert shown in future live streams.
- Animal behavior - Machine learning algorithm should be able to identify some animal behavior and notify the party responsible.

See Appendix B, Table 2, for a detailed list of validation scenarios.

2.12 - Functional Testing Plan

Throughout the past semester, we've had the opportunity to develop a functional test plan for the system we are building. While we have been testing during the second half of the semester, we decided to create a concrete plan to follow for the remainder of the semester. Since we do not know the progress that will be made over the summer, we intentionally left out the fall semester from our plan. Once we get back in the summer, we will evaluate the progress made and make an amendment to our project plan with the updated test cases we anticipate testing.

Test Risks / Issues

There may be risks or issues that arise as we are testing our 360 camera system. The primary issues that may arise are:

- Hardware: As with most project which require a hardware component, there is the potential that either the hardware doesn't work as intended. Should the hardware encounter any issues, we will make every attempt to fix the issue that we can think of. If the hardware is still having issues, we will seek advice and guidance from our faculty advisor, Dr. Duwe.
- Software: Our project is primarily a software project, where there are multiple components we need to connect. While we don't anticipate there being any risks associated with the software, there is always the potential for something to go wrong. Should we encounter any issues with the software, we will attempt to back trace our steps and figure out what the issue may be. If we are still stuck, we will reach out to our faculty advisor for guidance.
- Financial: While most of our project is utilizing open source software, there are components which if we are not careful can be a financial burden on True 360. For example, if our AWS account is improperly managed and we end up with a large bill of expenses, that's something that will need to be considered. Another example is if during the testing of the camera, one of the lenses is damaged, the camera may need to be replaced.
- Environmental: At the moment, there aren't many environmental risks to our project. Once the camera is placed in an actual exhibit, there may be environmental concerns to be aware of. One could argue that these are outside the scope of our project, but it's important to consider the bigger picture.

Items to be Tested / Not Tested

See Appendix B, Table 3, for a detailed list of items to be tested / not tested.

Test Regulatory / Mandate Criteria
While we don't anticipate many regulatory or mandate issues with this project, there is a potential for some to come up. For example, if a zoo has specific rules regarding running power cable through exhibits or security of the webcam, we may have to account that during our initial testing. While this seems like an edge case, this is certainly a possibility that we need to, at the very least, be thinking about.

There will likely be other regulatory or mandate issues that come up, but those will start to come up as the product is being installed in various zoo exhibits. As our criteria changes for these regulatory or mandates, we will also change our test plan to match them.

Test Pass / Fail Criteria
For each of the tests that we attempt, we will go back to our original project requirements and ensure that it meets the relevant requirements. While not every test has a relevant requirement, we want to verify that what we're building actually works as originally designed.

To ensure that the product meets and / or exceeds the product requirements, we must match each test to relevant project requirements which will help us verify that the design is correct and meets the clients needs.

Test Deliverables
As part of our testing plan, we will be creating a document which will be continually updated with the different tests we're performing. This document will be available to future developers and located in a shared place.

2.13 - Standards

During the design, development, testing, and maintenance phases, our team must adhere to standards in categories such as protocols, design patterns, and documentation. By adhering to these standards, we can ensure that our system will meet industry expectations of safety, security, maintenance, and professionalism.

Many of the project's hardware and software components will be developed around standard communication and connectivity protocols such as IEEE 802 MAC (Medium Access Control), IEEE 802.3 (Wired Ethernet), and IEEE 802.11 (Wireless Ethernet). Each of these protocols is incredibly important to our project, so it's also important that we follow the standards surrounding them. Our team has considered many of these standards and how they affect our project.

When implementing the system's software components, our team will follow the code modularity and readability practices which are relevant to the chosen programming languages. Being that our project utilizes various third-party libraries, APIs, and SDKs, we will follow the coding standards and other official specifications for those outside resources. Our code is also available on GitLab, so that if a change occurs we can approve or decline it. This is also how we can stay up to date with what everyone is working on.

In regard to our documentation, we have strived to ensure that all our documents have roughly the same layout and are well organized. Our team has organized the documents in such a way that if someone else were to pick up the project, they could easily get caught up within a few hours. This is entirely by design, due to one of our potential risks - a team member considering dropping the course. Should we have a team member that decides to drop the course, and we get a new team member, they will be able to get caught up and contribute efficiently.

Standard Protocols

Programming Practices

- Modular, well-commented code
- Utilize change management software such as Git

Documentation Standards

- Common theme for all documents
- Documented code and Git history
- All documents stored in central location, available to everyone on team

Ethical Practices

Based on our work so far, we've been able to follow the ethical practices for IEEE. Our team has done very well in terms of working ethically. Some examples of this are;

- We communicate frequently and discuss issues as they arise,
- Communicate any major issues with client as soon as possible,
- Ensure we're all aware of safety issues outlined earlier in the Project Plan.

IEEE Standards and True 360

Throughout our working relationship with True 360, we've learned a lot about IEEE standards and how they apply to our project. We've identified a few specific areas we're applying IEEE Standards;

Software Quality Assurance Processes

- Defined test cases
- Defined validation scenarios
- Change management software which ensures that when code is changed, multiple people much approve of the changes

Systems and Software Lifecycle management

- Avoiding short-term solutions and having a future-thinking mindset
- Writing quality documentation and code comments so whoever works on the project next will be able to catch up quickly
- Understanding that the technology is fairly new and working through a lot of technical issues

Requirements for Managers about Documentation

- Descriptive documentation for both the Project Plan and Design Document
- Regularly discuss plans and team updates
- Document weekly status reports

3 - Estimated Resources and Project Timeline

3.01 - Personnel Effort Requirements

See Appendix B, Table 4, for a detailed list of tasks.

3.02 - Other Resource Requirements

See Appendix B, Table 5, for a detailed list of resource requirements.

3.03 - Financial Requirements

The financial requirements associated with our project consist of purchasing hardware components and software services.

- Hardware
    - Garmin VIRB 360
    - Ricoh Theta V
    - *Insta360 Pro (purchased prior to the start of our project)*
- Software
    - Amazon Web Services (AWS) or other cloud storage solutions
    - Pusher Websockets service that will be used for communications between Admin Portal and the camera controller. This is a scalable service and pricing depends on number of connections and messages.

3.04 - Project Timeline

- Project Plan & Design Document
    - This consists of the completion of this document and any other UML design documents that we create for the project.
- Prototype version 1.0
    - At this point, we will begin our Agile Development process. It will consist of daily scrum stand-ups and twice a week in person meetings.
    - We will integrate the Continuous Integration & Continous Development tools provided by GitLab. The prototype will lay down the foundation required for a functionally complete system. The outcome will be all the necessary UI, backend, and hardware functioning to show that the project is feasible and that our approach will work.
    - All external resources (AWS/Server/Database Resources/etc..) will also be part of this prototype to ensure that the main components can communicate with each other and that we don't run into roadblocks later on with choosing the wrong technologies.
- Analyze prototype v1
    - This will be part of our in person meetings, we will focus on analyzing the prototype as we go along and fail early. This process will start a week before we develop the prototype, so that we are able to address any minor fixes quickly

the last week and perhaps find solutions for the problems. That will save us precious time when we begin development of Prototype version 2.0

- Prototype version 2.0
  - This will be main prototype that we will focus on this semester. The goal for this is to have a solid foundation of our design & a clear understanding of what the feasible solutions will be for the project next semester. It will also help us discard any technologies that will not work and identify any issues with the project design early in the development.
  - The deliverable of this will consist of a fully implemented UI (With placeholders on non-functional requirements). The UI will also have access an a Web API with reliable a realible Wiki for the API. Our API and backend will work completely and connect with any external resources, authentication providers, and with any webcam controls that are functional requirements.
- Analyze prototype v2
  - Similar to the previous prototype, analyzing this one will also be something we start doing early, while it's still in development.
  - The key outcome of this task is that we will have all the requirements to create our Final Project Plan, Software Architecture diagrams, and any design documents that we will need to begin development of the final product next semester.
- Next Semester
  - Sprints 4-12
    - Starting next semester will be reviewing any documentation that we drafted to both update it and to refresh ourselves with the project design.
    - In the beginning, the biggest priority will be to implement the project on a Cloud Based product
    - Since the project will have gotten larger and more functional, there will be a bigger emphasis on Bug Fixes and testing. We'll assign the first week of each sprint to identify bugs found during the last one and to fix them as quickly as possible.
    - As we near the completion of the Live Streaming & Processing tasks, we will begin working on the Machine Learning part of the project. Following this plan, we'll have a sprint dedicated to the Design Documents for Machine Learning based on our Live Streaming & Processing implementation. We'll next switch gears mostly into Machine Learning as we near the end of the semester.
    - Lastly, we'll allot the last week for all Hotfixes and a final Production Deployment as a deliverable to our Client.

See Appendix A, Figure 4 for a graphical representation (Gantt Chart) of our project plan.

3.05 - Foreseen Challenges

Camera Controller
This part of the project is currently under development, and is currently undergoing a lot of problems. We wanted to use a Raspberry Pi to handle the transcoding of the live stream, but it became apparent that it is not possible because Ricoh does not provide linux drivers for their camera (Raspberry Pi runs a linux distro). We are currently looking for another solution such as a board that runs a full windows 10 OS, or use Raspberry Pi as an access point to communicate with the camera. Moreover, we have ordered the Garmin Virb since it does not require drivers to stream from.

Embedding Logos
Since the live stream will be in 360° format, embedding 2D logos will require kind of process to transform logo to appear 2D on live stream.

Incorporating Machine Learning
The purpose of the machine learning algorithm as of now, is to determine the activity of animals in a given enclosure. When enough data is collected, a graph will be created to represent that activity in relation to time. Since the field of view is 360°, other movement not made by the animal might be added to the activity graph. This is ofcourse is undesirable, and the algorithm should be trained to only track the activity of the animal.

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

Appendix A: Figures

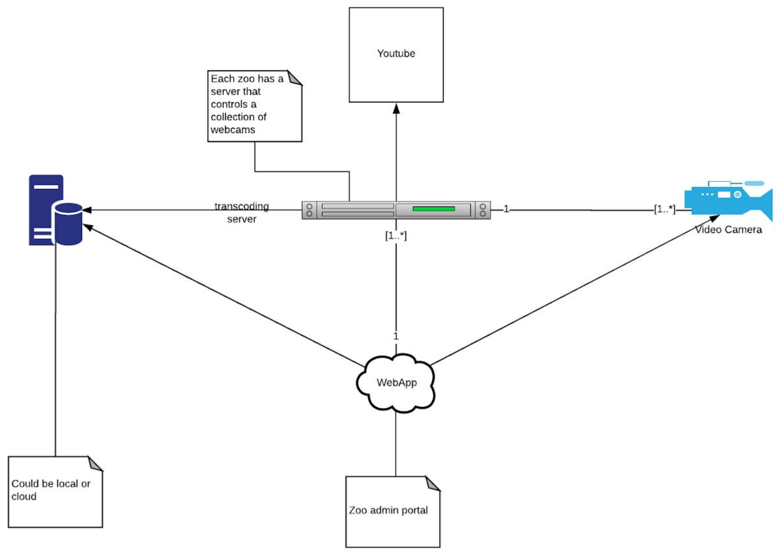

Figure 1 - Localized Design Approach

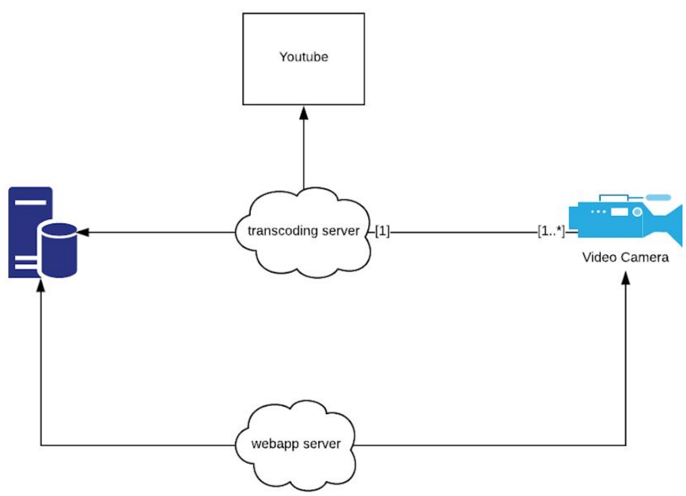

Figure 2 - Cloud Design Approach

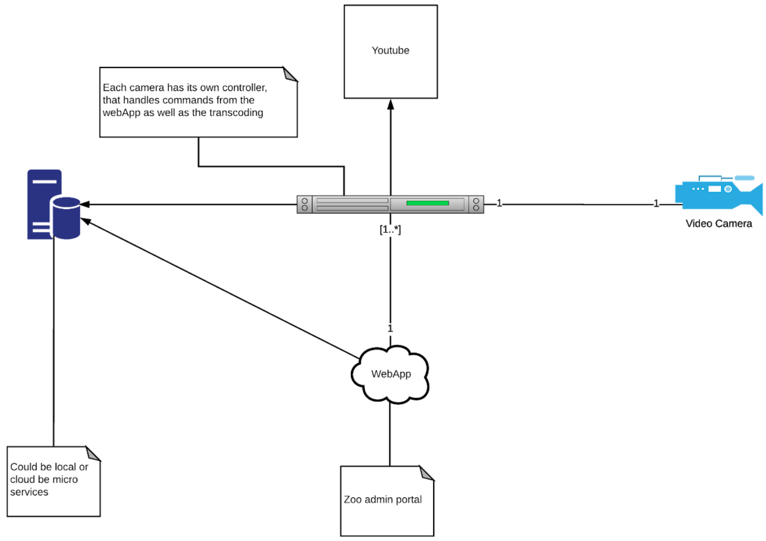

Figure 3 - Controller Design Approach

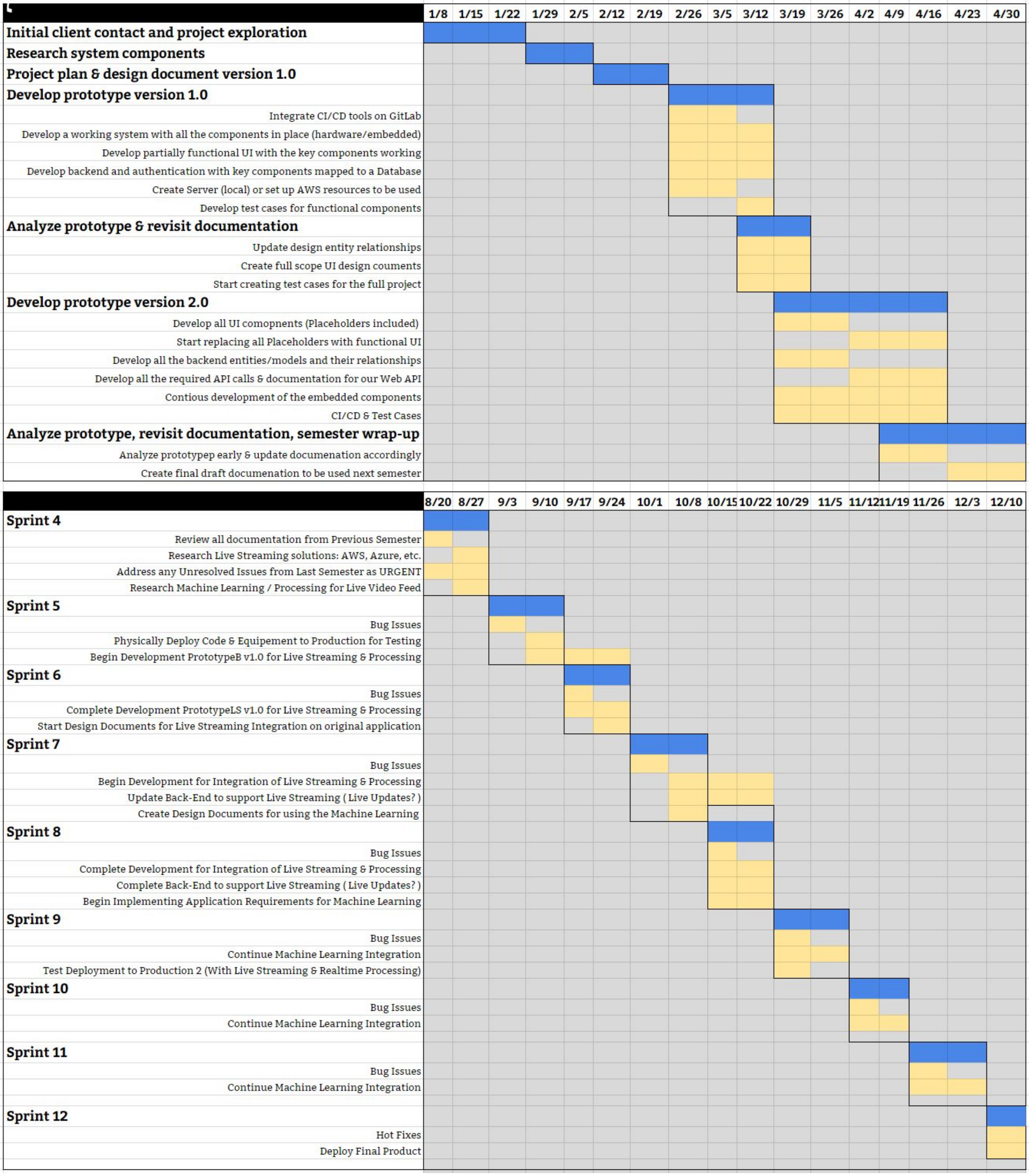

Figure 4 - Project Timeline

Appendix B: Tables

Test Case	Relative Milestone	Expected Result
TC-1	Milestone 0	Final product should be able to run for prolonged periods with no issues or interrupts
TC-2	Milestone 1	WebApp should include all necessary functionality to setup and controller webcams; start and stop streaming
TC-3	Milestone 2	Be Able to use WebSockets (Pusher) to deliver commands from the Admin Portal to the webcam controller
TC-4	Milestone 3	Zoo storage should be able to hold/store a desired amount of hours for each webcam.
TC-5	Milestone 4	Integrate desired logos to a live stream in an intuitive way, and in the correct location on the screen
TC-6	Milestone 5	Give activity graphs based on movements of animals
TC-7	Milestone 6	Suggest clips from exhibits where an animal was most active
TC-8	Milestone 7	Camera works as intended and the zoo is able to utilize the minimal viable product.

Table 1 - List of Test Cases

Validation Step	Relative Result	Validation Scenario	Expected Behaviour
VS-1	Camera Controller Transcoding server	Limit bandwidth available to the camera controller	Live stream quality should adjust based on the upstream speeds of the zoo's internet and receive and send commands in real-time
VS-2	Camera Controller	Unexpectedly interrupt the internet connection then establishing a connection	Camera Controller software should be able to re-establish a connection to the backend and continue the previously running process
VS-3	Zoo Admin Portal	Zoo IT staff wants to manage cameras which are difficult to physically reach	Webcams that are located far from the router or underwater should be reachable through the Admin web application.
VS-4	Video storage	Zoo curators want to be able to find a specific time when an animal was acting strange	Previous streams are retrievable if they are within the set timeframe. Streams not within the timeframe are deleted to give space to more recent ones.
VS-5	Logo Integration	Zoo marketing director wants to put sponsor logo at bottom of 360 video	Logo should appear 2D and deformed within the 360 video.
VS-6	Marketing App	Zoo marketing director will be able to manage marketing material from brands and overlay their material on video	A queue of sponsors will be accessible to zoo staff with any other information provided by the potential sponsor.

		stream	
VS-7	Animal behavior	Animal is acting abnormally, zoo curators are then notified	Algorithm should be able to differentiate between animal and other objects, and only reflect the activity of the animal on the graph.

Table 2 - List of Validation Scenarios

Item to Test	Test Description	Pass Requirements	Test Date	Team Responsible
Functionality at different network speeds	Camera controller should be able to stream in variable resolution depending on the network speed	Restrict bandwidth available to the camera controller, and check whether video quality was adjusted	5/1/2018	Embedded
Archiving & Receiving data from AWS	Simulate recording a video and upload to AWS, & simulate receiving the data from AWS to be presented on the web page.	Our system should be able to send and retrieve information from AWS.	5/1/2018	Embedded
"N" amount of cameras connected to the network	Use multiple cameras to test the functionality of the system with more than one camera.	System should be able to stream and archive at least two cameras simultaneously.	5/1/2018	Backend
Power to camera controller disconnects	Simulating a sudden interrupt with the flow of	Camera controller should be able to restart and	5/1/2018	Embedded

unexpectedly	power to the camera by disconnecting the power cable going to either the camera, server, or Raspberry Pi.	start up the camera controller software		
Internet disconnects unexpectedly	Simulating a sudden interrupt in the connection by either removing the ethernet cable or restarting the wifi.	Camera controller should be able to restart the communication interface and accept any incoming commands	5/1/2018	Embedded
API server crashes	If the API server crashes & is forced to restart. (AWS handles restart on crashes)	API server will be up and ready automatically on restart.	4/12/2018 - Passed	Backend

Table 3 - Items to be Tested / Not Tested

Task ID	Task Description	Task Priority	Projected Effort Required
001	Decide on a design architecture	High	4 weeks
002	Design a camera controller	High	4 weeks
003	Design a storage service	High	3 weeks
004	Embed logos in live streams	High	1 week
005	Design an admin portal (webApp)	High	2 weeks
006	Incorporate machine learning	Low	8 weeks
007	Design an upsell platform	Medium	2 weeks

Table 4 - List of Tasks

Resource Name	Resource Type	Description	Cost	Source
Insta360 Pro	Hardware	Professional 4K 360° webcam	~$4,000 (retail)	Purchased by client.
Garmin VIRB 360	Hardware	Rugged 4K 360° webcam	~$800 (retail)	Purchased by client.
Ricoh Theta V	Hardware	Portable 4K 360° webcam	~$400 (retail)	Purchased by client.
Raspberry Pi 3	Hardware	Board	~$35(retail)	Purchased by the client if needed.
Pusher	SDK	WebSocket Service	Free tier	Purchased by client if free tier is exceeded

Table 5 - List of Resources

4.04 - Changelog

<u>Note:</u> *Changes are listed in descending order (newest on top).*

- **2018.04.24 (Version 3.0.0)** – Final draft submitted
- **2018.03.25 (Version 2.0.0)** – Second draft submitted
- **2018.02.11 (Version 1.0.0)** – First draft submitted