

5-3-2016

# Augmented Reality in the Classroom


Patrick JB Foster

Loyola Marymount University, pfoster3@lion.lmu.edu

Sean Cunniff

Loyola Marymount University, scunniff@lion.lmu.edu

Follow this and additional works at: <http://digitalcommons.lmu.edu/honors-thesis>

 Part of the [Computer Engineering Commons](#), and the [Graphics and Human Computer Interfaces Commons](#)

---

## Recommended Citation

Foster, Patrick JB and Cunniff, Sean, "Augmented Reality in the Classroom" (2016). *Honors Thesis*. 122.  
<http://digitalcommons.lmu.edu/honors-thesis/122>

This Honors Thesis is brought to you for free and open access by the Honors Program at Digital Commons @ Loyola Marymount University and Loyola Law School. It has been accepted for inclusion in Honors Thesis by an authorized administrator of Digital Commons@Loyola Marymount University and Loyola Law School. For more information, please contact [digitalcommons@lmu.edu](mailto:digitalcommons@lmu.edu).



# PROGRAMMABLE, INEXPENSIVE VISUAL AID SYSTEM IN THE CLASSROOM USING AUGMENTED REALITY

A thesis submitted in partial satisfaction  
of the requirements of the University Honors Program  
of Loyola Marymount University

by

**Patrick Foster**

**May 5, 2016**

# PROGRAMMABLE, INEXPENSIVE VISUAL AID SYSTEM IN THE CLASSROOM USING AUGMENTED REALITY

Frank R. Seaver College of Science and Engineering  
Department of Electrical Engineering and Computer Science

**Instructor:** Dr. Hossein Asghari

**Contact:** [Mohammadhossein.Asghari@lmu.edu](mailto:Mohammadhossein.Asghari@lmu.edu)

**Team:** Patrick Foster, Sean Cunniff

**Date:** May 3, 2016



# Abstract:

According to the CDC, 14 million people in the United States have visual impairment, and 21.4% of them cannot see with the use of a glass lens [1]. These people are not necessarily blind, but are considered to have low vision. Low vision can have an exceptionally negative impact on a student's ability to learn, especially when subjected to the conventional education system. In this environment, students are expected to adhere to a lecture that delivers most information visually via a whiteboard or a projector screen. Various technologies exist to aid the visually impaired in the classroom, but all are limited to individual work. No current solution works for a lecture.

It is important to recognize, however, that not all information in an observable scene is necessary. This is evident in the designs of existing technologies for the visually impaired, which emphasize only crucial information such as text on a page. This selective processing can also be achieved in software with augmented reality. The goal of this project is to create a customizable application for a smartphone that implements this selective processing in order to make it easier for visually impaired students to engage with and learn from lectures.

This application will consist of two primary components working with each other. The first component is the image processor. The purpose of the image processor is to modify the image received by the camera on the phone in order to selectively alter key aspects of it. For example, in the context of this project, the image could be that of a whiteboard with handwritten notes on it. The processor could then emphasize the notes by detecting them and stretching them to fit the entire screen. This processor must work on a real-time feed in order to be useful for students in the classroom. For this reason, hardware acceleration, the other primary component, is needed. Hardware acceleration consists of parallelizing fundamental processes to speed up computations. This is achieved in practice by distributing these computations to the phone's GPU.

Specifically, this application is written in the Java language for the Android platform. The application uses OpenGL ES, a C-like language for the mobile platform, in order to perform the image processing. Filters written in OpenGL ES are used to modify the image read by the phone's camera. Using these filters, the application can modify an image by stretching, magnifying, and enhancing the color and contrast. The specific processes included in the application include Sobel Edge Detection, Dilation, Zoom and Contrast. The hardware acceleration is also performed using OpenGL ES.

The application was tested with the help of Jimmy Vivar, a sophomore Civil Engineering student at LMU with amblyopia (lazy eye) and retinitis pigmentosa. The combination of both of these conditions contributes to reduced peripheral vision. Jimmy's vision is not helped greatly by eyeglasses, and thus is a viable candidate for testing the application. He is crucial in defining the specific functionality of the application and verifying that it works correctly in ensuring that he can see the whiteboard.

A direct consequence of this project is solving a problem in the classroom for visually impaired students not yet addressed by current technologies. While this project was relatively unsuccessful in specifically aiding students with impairments like Jimmy's, this project may still be useful in applications for people with slightly less severe peripheral vision impairments.

# Table of Contents

1. Introduction:.....	4
1.1. Background: .....	4
1.2. Current Solutions: .....	4
2. Project Objective: .....	5
2.1. Problem Statement:.....	5
2.2. Customer Requirements: .....	5
3. Proposed Solution:.....	5
3.1. Formulation: .....	5
3.2. Technical Requirements and Constraints: .....	6
3.3. System Description:.....	6
3.4. Standards: .....	7
4. Experimental Test and Demonstration .....	8
4.1. Benchmark Testing: .....	8
4.2. Working Prototype and Development:.....	8
5. Ethical Considerations .....	10
6. Contribution to ABET Program and LMU Values.....	10
7. Conclusion:.....	11
8. Suggestions:.....	11
9. References: .....	12
10. Appendices: .....	13
10.1. Responsibilities of Team Members:.....	13

# 1. Introduction:

## 1.1. Background:

Approximately three million people in the United States have a form of visual impairment that cannot be corrected with simple light refraction [1]. In the literature, people with this degree of visual impairment are referred to as having “low vision” [2]. This can prove especially problematic in the classroom, because many of these people are students. However, not all information needs to be emphasized in an image. For example, in a classroom, text on a whiteboard may be the only crucial visual information. This selective processing lends itself directly to augmented reality.

Augmented reality is a live view of a real-world environment where the elements of the environment are augmented by software, such as video or audio. Augmented reality is not as well-known as virtual reality, where a simulated world is used instead of a real world. Today, augmented reality is said to have potential applications in education, workplaces, and operating rooms [3].

## 1.2. Current Solutions:

There are no current augmented reality solutions to visual impairment in the classroom. Other technologies for visually impaired students exist and are effective. These technologies usually provide high magnification or high-contrast for the user. For example, Freedom Scientific’s RUBY handheld video magnifiers are pocket or purse sized magnifiers capable of providing 2x to 13x magnification and 20 high-contrast color viewing modes. However, this magnification is unsuitable for use in lectures, as even with 13x magnification, the resulting image is too small. Additionally, the battery life of the product is 2 hours, and this is too short for some lectures [4]. Optelec produces a larger scale product called the Optelec ClearNote SD, which provides a magnified real-time view of locally held objects, such as books or paper. The device can provide up to 40x magnification and is portable, but cannot be used to view a lecture, and is instead only useful for the aforementioned locally held objects [5]. Similarly limited, but also used to aid the visually impaired, is high-contrast or thick-lined paper. Color enhancements, such as color filters or the use of specialized felt tip markers, also exist in the market. These alterations support the previous claim that selective processing works for helping visually impaired people see and learn.

However, none of these solutions are suitable for lectures because they are primarily focused on individual work present on the desk. The primary component of lecturing has yet to be addressed, and therefore real-time learning for the visually impaired has yet to be addressed by an augmented reality solution.

## **2. Project Objective:**

### **2.1. Problem Statement:**

This project is important because many students with visual impairments struggle in a classroom setting, where most information is delivered visually—via a whiteboard or presentation. This clearly has a drastic impact on students’ ability to learn, as they are immediately disadvantaged and have to work unnecessarily harder to keep up with the material. Additionally, this is problematic for the teacher, because the teacher must work with visually impaired students more to keep them up to speed.

The goal of this project is to develop an application that will enhance the classroom experience for students with visual impairments by making the lecture easier to see. The application should selectively emphasize certain information for the student, such as handwritten text or diagrams. This will be realized in the form of a dynamic filter that processes and modifies live video so that students can view a lecture in near real-time from the first or second row of a classroom.

This project would inherently enhance students’ ability to learn by making lectures easier to see. This project should extend a conventional education to a larger demographic. Furthermore, this aid can also extend to work and group situations outside of the classroom.

### **2.2. Customer Requirements:**

In order to determine customer requirements, Jimmy was consulted to specify his needs for the application. While the project is focused on unique impairments that will be individually accommodated for, the specific solutions for Jimmy’s particular impairments were not known.

## **3. Proposed Solution:**

### **3.1. Formulation:**

After much deliberation, it was decided that the Android platform would be used to develop the application. The Android platform uses the Java programming language in the Android Studio IDE. The team has some background experience in development which lended itself well to this solution. Additionally, OpenGL ES, a C-like language meant for interfacing with the mobile GPU, performs the image processing and hardware acceleration using Android NDK. Table I is the decision matrix used to arrive at this solution, which compares platform solutions between Android, iPhone, and the web.

TABLE I  
Decision Matrix Comparing Platform Solutions

Criteria	Weights	Solutions (Platforms)		
		iPhone	Android	Web
Experience	0.10	0.10	0.64	0.26
Development Time	0.56	0.10	0.64	0.26
Existing Libraries	0.10	0.26	0.64	0.10
Hardware Access	0.25	0.10	0.64	0.26
<b>SCORE</b>		<b>0.12</b>	<b>0.64</b>	<b>0.24</b>

### 3.2. Technical Requirements and Constraints:

From the team’s research, it was determined that the following minimum requirements must be met for the project to be considered successful:

- The average frames-per-second rate of the application must be above 16—the image must not be jittery for the user, otherwise it is too distracting to be useful. The application should be able to ideally reach 20fps for an optimal real-time feed.
- The phone selected for development must have a minimum resolution of 1280 x 720 (720p)—any resolution less than this is not high definition, and thus is inadequate for an immersive classroom experience. However, the resolution cannot exceed 1920 x 1080 (1080p) because the average fps rate requirement will not be met.
- The phone selected for development must be compatible with the selected headset, in order to ensure that the phone fits.
- The GPU must be compatible with OpenGL ES (any version) in order to be able to do hardware acceleration.
- The application must be able to operate within a range of 7-12 feet (2.1-3.7 meters) from a whiteboard or projector screen. This corresponds to the average of the distance between a whiteboard and the first and second row of a typical classroom.
- The application must provide user controls for adjusting the image to accommodate for unique visual needs, the details of which were determined through consultation with Jimmy.

### 3.3. System Description:

The application provides controls to allow the user to customize the filters used in the image processing. This instantiation is a one-time setting when first opening the application, but can be modified later by the user. The application then reads in an image from the mobile camera. This image is processed according to the aforementioned settings using filters written in OpenGL ES. Intermediate computations are parallelized and sent to the smartphone’s GPU also using OpenGL ES. The result of the computations are then sent back to the image processing functions and repeated until all processing is complete for the image via the GL thread. This particular functionality is shown in Fig. 2. Once the processing is complete, the processed image is scaled down to occupy half of the phone screen and duplicated, in order for both eyes to be able to see



the image. The resulting image is then displayed on the phone screen. This process, from reading the image from the camera to displaying the image on the screen, is repeated until the application is closed. This overall process is illustrated in Fig. 1.

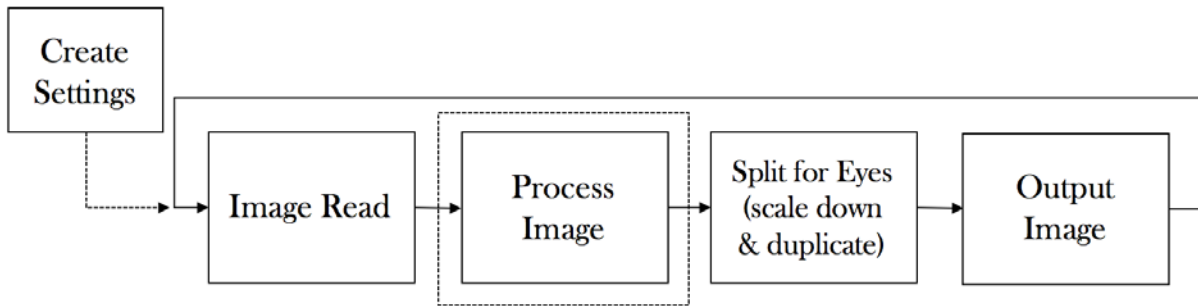


Fig. 1. High level system diagram flowchart

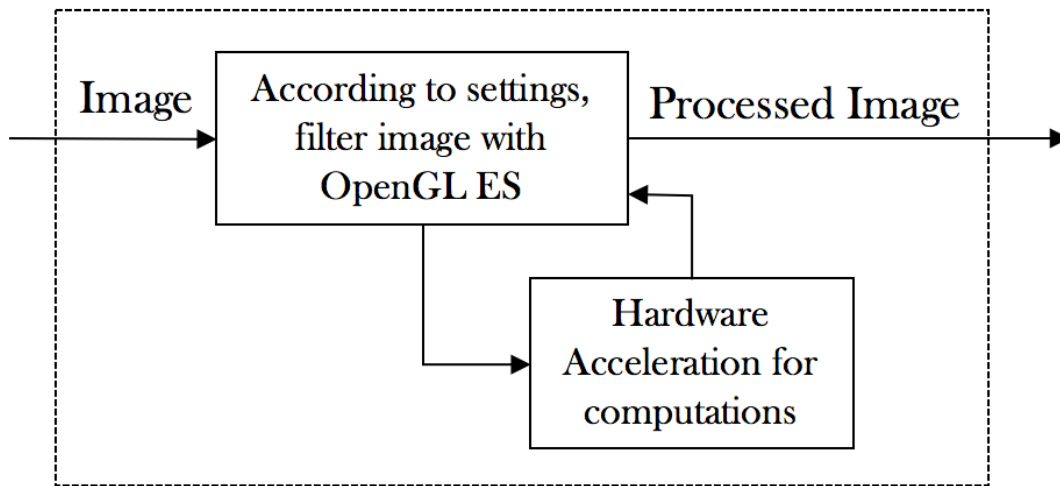


Fig. 2. Image processing diagram flowchart

### 3.4. Standards:

The application code will be compliant with Oracle’s “Code Conventions for the Java™ Programming Language” [6] in order to be syntactically correct. OpenGL ES will be used in order to adhere to industry graphics standards. Additionally, testing on the beta users will be performed in accordance with the NSF policy for human experimentation [7].

## 4. Experimental Test and Demonstration

### 4.1. Benchmark Testing:

Testing of the application was performed entirely using black-box methods by both the developers and by Jimmy. Stress tests were performed on the running application by pointing the camera of the phone at the whiteboard and simulating a typical use case for extended lengths of time by the developers. This was done in order to verify correct functionality of the working device and to also ensure that the framerate and imaging range were sufficient as outlined in §3.2. The framerate was varying between 4-7 fps for the second version (detailed in §4.2) of the application using OpenCV, but this was improved to real-time in the third and final version by switching to OpenGL ES for image processing and hardware acceleration. The imaging ranges were sufficient for both applications. Additionally, this testing aided in identifying a bug where more than one filter could not be cascaded.

Jimmy tested the third version of the application with the phone mounted in a Vigica headset, situated approximately 12 feet away from the whiteboard. The application was tested using a blue Expo marker and by drawing both common engineering phrases such as “Fourier Transform”, as well as various shapes to test if Jimmy was able to read the former and identify the latter. Jimmy did have limited success where he was able to vaguely identify shapes and some letters, but had more difficulty with words. Ideally, a user of the application would zoom in on a section of the whiteboard, apply the combined Sobel Edge Detection and Dilation filter that fit their vision appropriately, and be able to successfully read the whiteboard. However, this was more challenging for Jimmy due to his peripheral limitations. In his case, he struggled to find a consistent focal point to zoom into, and constantly ran into orientation issues with the whiteboard. This restricted the usability of the application for Jimmy.

### 4.2. Working Prototype and Development:

The first version of the project used an open-source camera application as a basis in order to understand the foundation for the Android platform, as well as hardware interfacing with the camera. There were significant limitations with the application, and most of the effort was spent on configuring Gradle as well as the Android emulator to test the application without physical hardware.

The second version of the application was primarily focused on image processing using OpenCV (available by configuring Android NDK to use native code written in C++). This particular version also used a split camera view to be used in a headset, as originally planned. Configuring Android NDK proved to be challenging because native code is still considered experimental by Android, and thus cannot be compiled by Gradle itself. This was first handled by external compilation of the isolated C++ files. Gradle was used separately to compile the rest of the application. This became a tedious process, and so efforts were made to automate the NDK compilation within Gradle. At this point, image processing was not utilizing any hardware acceleration. However, by using OpenCV in the native code, this bypassed the Android Java compiler and the image processing was compiled directly to hardware. A relatively tolerable

framerate was produced by the Probabilistic Hough Line Transform filter used in the C++ code. This version of edge detection searches the image for adjacent points and broadly determines whether or not a particular set of points corresponds to a straight edge. This filter then draws these valid set of points to the screen over the original frame. It is worth noting that this filter implements a subset of two sequential stages (Canny Edge Detection and the Hough Line Transform itself), and thus introduces inherent lag. In order to improve the framerate, hardware acceleration was needed. However, the OpenCV libraries could only be directly integrated with OpenCL or CUDA, and the phones used for testing only work with OpenGL. Therefore, OpenCV could no longer be used due to the speed required for usability of the application. Additionally, significant tweaking was needed for the filtering used, as it only sporadically worked on straight lines and had major problems with curves. This version of the application can be downloaded at [https://github.com/patrick-foster/AR\\_Project](https://github.com/patrick-foster/AR_Project).

The third and final version of the application was a fundamental restructuring of the Android framework as well as the image processing used by the previous application. OpenGL ES was used in lieu of OpenCV due to its superior speed and quality, largely due to the hardware acceleration inherent in the code. The implementation in Android of OpenGL ES significantly differs from OpenCV, because Android requires functions to be passed in as strings to abstracted GLES20 calls within Java. These strings are written in proper OpenGL ES code, which specifies rules for affecting a localized group of pixels in order to process the region. This ruleset is repeated for all portions of the image from the camera hardware by a GL thread running in the background of the application. The GL functions for this application implement two parts of the OpenGL pipeline, which individually specify the regions of the image that are to be affected by the processing and the processing itself. These respectively are the vertex shader, representing the coordinates within the space, and the fragment shader, representing the aforementioned ruleset associated with the vertex shader. Additionally, many features were added to the application including pinch zoom, multiple cascaded filters created to meet customer requirements (including Sobel Edge Detection, Dilation, and Contrast), adjustment controls for the cascaded filters, and a cleaner user interface. Sobel Edge Detection convolves two 3x3 matrices of x- and y-derivatives for a particular pixel region with the original pixel region in order to determine the difference in depth intensities between pixels. It then finds the magnitude of the gradient approximation of the convolutions, outputs to an altered Preview buffer, and then to the screen. This form of edge detection requires only one stage of processing, as opposed to the aforementioned Probabilistic Hough Line Transform because the innate result of the convolutions is the processed image. Cascaded filters were achieved by utilizing the GPUImageFilterGroup class to build a List data structure that consists of multiple GPUImageFilters. In this case, the output of Sobel Edge Detection is piped in to the Dilation filter, which is important so that Dilation is only performed on the highlighted edges of the image. The Dilation filter finds the maximum intensity of the red channel for a 3x3 pixel region, and applies this intensity to all pixels within that region. This essentially behaves like a low pass filter to the image, removing small spaces between highlighted edges. Additionally, the aspect ratio of the image was corrected from the originally distorted view in the second version of the application, which was almost a 1:1 ratio. This version of the application can be downloaded at [https://github.com/patrick-foster/AR\\_Project\\_gpuimage](https://github.com/patrick-foster/AR_Project_gpuimage).

## 5. Ethical Considerations

The most important ethical consideration in this project was ensuring that testing on Jimmy, while non-invasive in nature, was compliant with the NSF standards for human experimentation, and any other ethical review board pertaining to human experimentation. This is to ensure that the team adheres to the IEEE Code of Ethics, specifically “to accept responsibility in making decisions consistent with the safety, health and welfare of the public” from contention 1 [8]. Furthermore, contention 5 of the Code of Ethics addresses the impact of technology and its consequences on society. While this application may not be far reaching in terms of its consequences in a school environment, the team acknowledges that the application could be used in contexts outside of the classroom, such as a work or group environment. Therefore, the team designed the application to work within suitable bounds, regardless of the intended environment.

## 6. Contribution to ABET Program and LMU Values

The team undertook this project in order to cultivate LMU’s Core Values, specifically that of Service to Others—“... a culture of service in which we apply our knowledge and skills to better the human condition” [9]. This project specifically attempts to improve vision for students in a classroom environment, and thus improve the quality of life for fellow students.

Additionally, the team strove to affirm the Core Value of Academic Excellence—“...an uncompromising standard of excellence in teaching, learning, creativity and... to stretch their intellectual boundaries through diverse experiences of engaged, rigorous, critical, expansive and transformative learning” [9]. In undertaking this project, the team acknowledged the difficulty and potential value of accomplishing the task at hand.

With respect to the ABET Program, this project specifically addresses learning outcomes (a),(c), (e)-(g), and (k) [10]. (a) is addressed through the application of engineering to this project, both in methodologies and in practices (in particular, software engineering). The entire project is a design of a system for a real-world need, and therefore addresses (c). The team addresses (e) through the identification of engineering problems, the solution of which forms the basis of the end product. This project has real ethical concerns associated with it in the reliance of Jimmy to verify functionality of the product, which addresses (f). (g) is addressed through communication with Jimmy, communication within the team, and communication with LMU faculty and staff. The team addresses (k) through the use of various libraries, platforms, and tools to provide a software solution to the underlying problem of the project.

## 7. Conclusion:

This project consisted of developing an application to make conventional lectures more accessible for students with visual impairments. The application was written in Java for the Android platform. It ultimately utilized OpenGL ES for both image processing and hardware acceleration. This project took the entire semester to complete fully. There were approximately three stages to the process of realizing the final application. The first stage was acquainting with the Android platform and conventional architecture. The second stage was learning the image processing using OpenCV libraries implemented in native C++ code, and the corresponding compilation of said code. The last stage was rewriting the image processing and incorporating hardware acceleration with OpenGL ES, and adding features to meet customer requirements. This project is compliant with NSF, IEEE, LMU standards and values, and addresses six ABET outcomes. While there were limitations when testing with Jimmy Vivar, the application may have use for users with less severe peripheral vision limitations. Additionally, this application could be improved for users like Jimmy, as outlined in the Suggestions section. With tweaking, this application can be used in group and work situations beyond the classroom.

## 8. Suggestions:

There are two categories of suggestions for this project. The first is retrospective, and the latter is for future improvements. For the retrospective suggestions, the most notable is that there were severe limitations imposed by choosing a Samsung Galaxy Note 3 phone for testing. This phone requires OpenGL ES and a separate application to prevent automatic locking when running the application. Had a different Android phone been selected that is CUDA compatible, OpenCV would still be in use with the final application as it is directly compatible with CUDA hardware acceleration. This would have minimized bottlenecks introduced by Android interfaces, and would have reduced developer time spent restructuring the application.

For future improvements, one of the most notable features missing from the application is external control of the application through the headset. This would involve establishing a magnetic listener in Android to check for magnetic field disturbances caused by the headset and sensed by the phone. The application would then affect the zoom of the camera accordingly. This is needed because finding a focal point on the whiteboard for users is very cumbersome without external zoom controls. Another improvement would be to add external controls for Dilation adjustment. Currently there are five different versions of Dilation (degrees of intensity) that are used by the application, where each version corresponds to a different filter. Ideally this could be an adjustable parameter that the user can modify. Additionally, the slider corresponding to the variability of the line width for the Sobel Edge Detection could be made more sensitive and accessible via the headset. Unfortunately, only one of these improvements can be chosen because there is only one external control on the headset. Further research and work with potential candidates would be needed to determine which controls would be most optimal for meeting customer requirements. The final improvement determined by the team is an auto-hide feature for the slider and filter button in order to minimize obstruction of view.

## 9. References:

- [1] Centers for Disease Control and Prevention. (2015, September 30). *National Data* [Online]. Available: <http://www.cdc.gov/visionhealth/data/national.htm>.
- [2] P. Cox and M. Dykes, "Effective Classroom Adaptations for Students with Visual Impairments," in *Teaching Exceptional Children*, vol. 33, no. 6, p. 68, 2001.
- [3] A. Kofman. (2015, June 9). *Dueling Realities* [Online]. Available: <http://www.theatlantic.com/technology/archive/2015/06/dueling-realities/395126/>
- [4] Freedom Scientific. *RUBY Family of Products* [Online]. Available: <http://www.freedomscientific.com/Products/LowVision/RubyProductFamily>
- [5] Optelec. (2013). *Optelec ClearNote SD* [Online]. Available: <http://ctech.lowvision.com/optelec-clearnote-sd/>
- [6] *Code Conventions for the Java™ Programming Language*, 1999.
- [7] National Science Foundation. (1991, June 18). *45 CFR Part 690: Federal Policy for the Protection of Human Subjects* [Online]. Available: <http://www.nsf.gov/bfa/dias/policy/docs/45cfr690.pdf>
- [8] IEEE. (2015, November 23). *IEEE Policies - 7.8 IEEE Code of Ethics* [Online]. Available: [https://www.ieee.org/documents/ieee\\_policies.pdf](https://www.ieee.org/documents/ieee_policies.pdf)
- [9] Loyola Marymount University. (2004). *Core Values* [Online]. Available: <http://academics.lmu.edu/strategicplan/strategicplan2012-2020/strategicplantableofcontents/corevalues/>
- [10] ABET. (2014, November 1). *Criteria for Accrediting Engineering Programs* [Online]. Available: <http://www.abet.org/wp-content/uploads/2015/05/E001-15-16-EAC-Criteria-03-10-15.pdf>

## **10. Appendices:**

### **10.1. Responsibilities of Team Members:**

As the team consists of only two people, and the project consists of components that are too complex to isolate solely for one person, the work was split evenly for each component in the system. The tasks themselves were assigned by both members using Slack, which also doubles as a convenient means of communication between the team.