User Information Augmentation. Vision Based Information Delivery System

Project Plan

Group: sddec19-14

Client: Radek Kornicki

Advisor: Aleksandar Dogandžić

Team members:
Omar Abbas
Jonah Bartz
Aaron Michael
Dennis Xu

Team email: sddec19-14@iastate.edu

Team Website: http://sddec19-14.sd.ece.iastate.edu

Version 2
4/24/2019

Table of Content:

List of Figures:

List of Tables:

List of Definitions:

HUD - Heads up Display
GUI - Graphical User Interface
HMI - Human Machine Interface/Interaction
NFR - Non-functional requirement
FR - Functional Requirement
UART - Universal Asynchronous Receiver/Transmitter
USB - Universal Serial Bus
TobiiSDK - Software development kit used for the tobii eye tracker 4c
GazeSDK - Software development kit used for the Android devices use with the tobii eye tracker 4c
CPU - Central Processing Unit
GPU - Graphics Processing Unit
ARM Processor - An ARM processor is one of a family of CPUs based on the RISC (reduced instruction set computer) architecture developed by Advanced RISC Machines (ARM) (Bigelow 1)
X86 processor - A CISC (complex instruction set computer) based computer architecture that was developed by Intel and AMD
ROS wrapper - Robotic operating system used for drivers
GPIO - General purpose Input Output
SSH - Secure shell protocol used for communicating with the shell of machine remotely

1. Introductory materials

1.1 Acknowledgement

We would like to first thank Mr. Radek Kornicki for providing supplies and guidance for the project. He has been providing us with all of the materials that we need to pursue this project. We also want to thank Dr. Aleksandar Dogandzic for supporting us with advice about how to develop our project. This experience has been made possible to us thanks to the Iowa State University ECPE Program and its cooperation with the enterprise corporation Danfoss Power Solutions.

In addition, the Virtual Reality Center on Iowa State's campus has provided support by showing their virtual environment. While augmentation is different from virtualization, there are many similarities between the two subjects, and the Virtual Reality Center helped by providing support about the eye tracking device. Over the semester, more support has been provided by the University of Pennsylvania.

1.2 Problem Statement

There is a lot of danger when it comes to operating heavy machinery. These machines can do a lot of damage to people and property if not carefully used. In many cases, the user has a hard time knowing what is happening around them. This, in conjunction with a distracted user, can provide a very dangerous environment to work in. Mr. Radek Kornicki from Danfoss wants to determine the viability of a user information augmentation system in improving safety when operating heavy machinery.

Mr. Kornicki wants to test how feasible a heads-up-display (HUD) system is to increase safety. The system will show visuals about the environment while the user is operating heavy machinery. To accomplish this, we want to make the windshield of the vehicle an HUD. This HUD will give information to the user about potentially dangerous situations that they might be in. This system will indicate things that the user should be looking at and also determine whether or not the user is doing something dangerous.

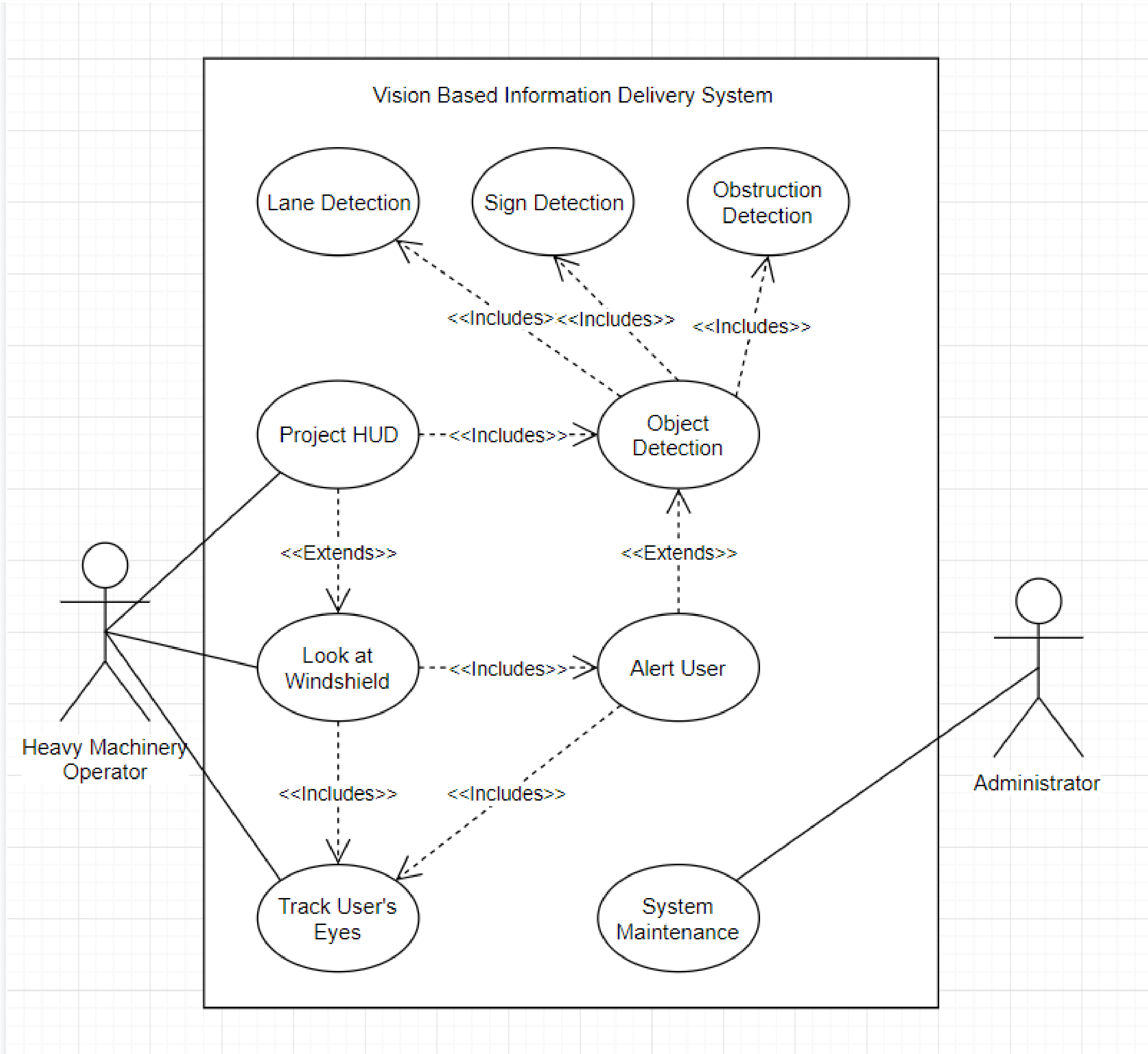

Figure 1: Use Case Diagram.

1.3 Operating Environment

Our client is the The Danfoss Group, known for manufacturing products and providing services used in powering heavy machinery, solar enginery refinement, food preservation and much more. The R&D division of Danfoss is determined to improve the technology in off-highway vehicles including agriculture and construction. Our embedded system may be used in any of these branches and given their agricultural focus we may safely assume that this can be put inside a vehicle that is moving in a muddy terrain or other less than ideal conditions. Hence the actual module itself should not be exposed but the cameras monitoring around the proximity of such a vehicle may in fact be affected. In our design the cameras will be separate and if this is the case, they should not be exposed to extreme heat, wet or other hazardous conditions for cameras. The module itself is 5 cm by 9 cm but should be kept stationary away from hazardous elements.

1.4 Intended user(s) and Intended use(s)

The intended users for our solution are heavy machinery operators. Heavy machinery operators are workers who using heavy machinery (like combine harvesters, cranes, and bulldozers). Because of the nature of heavy machinery, we infer that the operators are trained on specific vehicles but we can't assume that they have a lot of experience with software technologies. This means that we need to design a system that is easy to set up and easy to use for a range of vehicles. The images we draw onto the screen must also be clear and concise. This is because our product is targeting people in a busy work environment.

The intended use for our solution is improving safety when someone is operating heavy machinery. The solution needs to be fairly generic so it can be put into a lot of different machines without too much modification. That being said, there is a large diversity in heavy machinery with very different requirements. This means that we also need to make it easy to expand and iterate on the solution.

For the non-industrial user, a slimmer version of this project can be used for cruise control or long distance driving. The object detection system may alert an autonomous vehicle, signalling it to halt indefinitely. The eye tracker could monitor the eyes of the driver and halt the car if it sees that they may have fallen asleep.

1.5 Assumptions and Limitations

As a response to conditions referenced to us by our client, we are to make the following assumptions:

- The module will not be used in a turbulent situation.
- Will be operated in an environment where power is provided.
- Will only have one user at a time.
- For the prototype, information will be projected onto a screen or windshield.

To justify these assumptions, turbulence may not be ideal in the situation where we are parsing video input as the feed may be shaky. We are also making the assumption that this module will be operated in an environment it will be connected to a power source. A battery may be installed later but for the purposes of project, the prototype is meant for stationary means and not mobile. Finally, the eye tracker will be calibrated for one person at a time, and as such, we are assuming that is the maximum number of users.

Below is a list of the limitations we are under:

- Tests on the module will be undergone in a stationary, rather than mobile environment
- Less support as x86 - Arm conflict is not commonly addressed online
- Will only have one user at a time
- Data projection must be 70% transparent as stated under Iowa windshield laws

In addition, since we are working on a college campus, we are limited to testing the system in a stationary condition. Until we have permitted to use the system in a mobile situation provided on or off campus, we will not be able to test its efficiency on the move.

1.6 Expected End Product and Other Deliverables

Our expected end product is to design and build an eye triggered interface that will assist in using heavy machinery and electronic applications. This interface will exchange information with the user regarding the environment they are in to give them better intuition with their technology and to tighten the grip between the user - machine interface. We are to deliver a prototype consisting of the aforementioned components that accomplishes the description as efficiently as possible.

Our final deliverable will be the system composed of the following items:

Hardware:

- Tobii Eye tracker
- NVidia Jetson TX2
- RGB and infrared cameras
- Projector

Software:

- Programmable framework to suit needs of the project incorporating Tobii/Gaze SDK
- Working lane detection and basic object detection of road signs
- Code for detecting if sensors are working
- Code for outputting frames for a HUD

2. Proposed Approach and Statement of Work

2.1 Objective of the Task

In modern industrial and agricultural areas there is definitely need for critical attention to detail. the objective of the task is to minimize the amount of error during that session by any means. An example can be confirming that someone looked at a certain spot in the screen to make sure that they know it is there. Certain things like this could make all the difference in production and minimizing accidents.

2.2 Functional Requirements

Able to identify lanes on the road: Lane detection is a basic, but important, improvement to safety for heavy machinery. Since these vehicles can be large and high up, being able to detect whether or not the driver is inside the lane would be very useful.

Able to identify important objects on the road (signs, people): This is another component that would significantly improve the safety for heavy machinery.

Able to track user's eyes in the vehicle: A system to track the users eye movement can tell the system whether or not the user is distracted.

Project this in a heads up display style: Since the environment can be very noisy we want to be able to give information and using a HUD is the best solution.

A system that will track eyesight over a large visual field: Given the range of vision that someone has, we want to be able to track the eyesight of the user in a large area.

Can determine whether a sensor has been compromised: Since our solution is meant to increase safety we need to make sure that the data we receive from the sensors are correct.

2.3 Constraints Considerations

A constraint that was predicted early on by one our colleagues is the fact that we have little to no support for ARM processors in the market. That being said, connecting the Tobii Eyetracker (An x86 architecture processor) to a Jetson TX2 (An ARM architecture processor) is a non-trivial task. It would require reverse engineering the drivers and rewriting them for an ARM processor to achieve this goal, so the first constraint we have is that we do not have direct communication between our two input devices. This constraint may burden us with quite a bit of latency.

Another constraint we must deal with is the fact that we are programming a real time system. Therefore, all of the calculations processing must occur simultaneously and algorithmically so as to conclude the project run within a timely manner. If it is not then we will run into the issue where the system is not alerting the user of potential issues fast enough and maybe taking some time recognizing objects.

Lastly, our object detection is limited to OpenCV. This demands that we provide data about our target, which can be a lot of work to do if we have a lot of targets. This is because OpenCV is only an image processing tool. If we use machine learning tools like Tensorflow, then we are able to train our project to detect the data we want it to, rather than hardcode the information ourselves. Yet it seemed that we were unable to install Tensorflow due to driver issues and dependencies. If we can get Tensorflow downloaded on the Nvidia Jetson than it could be beneficial to use; however, it's not worth spending too much time on as machine learning is slightly out of the scope for this project.

2.3.1 Relevant Standards

One of the major standards that we need to accommodate is the standards about how much a car windshield can be tinted, as shown in figure 2. What this means is that anything that we use to help project something onto the windshield, like projection film, can't tint it more than 70%. Another standard that applies to us is UART. UART is a standard describing how to transmit data from one device to another device. We need to use something like UART since our design will require us to transmit a small amount of data between two machines. For the same reasons, USB is a standard that is applicable to us. The connection between the two computers is likely going to be using USB while transmitting UART data.

2.4 Previous Work And Literature

Tobii, from whom we get our eye tracker, has their own work with eye tracking in the automotive field. Much like this project, Tobii seeks to watch the user's eyes and alert them when they are distracted, either drowsy or looking away from the road. Unlike our project, which uses a full size Tobii Eye Tracker 4C, Tobii's own automotive project uses the Tobii EyeChip. A pro of this approach is that the system can recognize different users and can interact with the car to change the seat position, mirrors, etc. which is beyond the scope of the proposed design in this paper. The cons of this approach is that it interacts with the dashboard of newer cars, meaning without a smart display system in the car, the user cannot use the alerts that the Tobii system would send to them. Our approach seeks to fix this problem by adding a HUD which can be installed into any existing vehicle, allowing for a broader user base than for the Tobii automotive system as it is in its current state.

Nuance, the company who created Gaze, a software that integrates with the eye tracker, has produced their own car which monitors user's eyes and displays alerts. In their implementation a tablet that runs windows to display notifications to the user when applicable. The pros of this approach compared to ours is that it implements microphones that allows the system to look up information for the user and either display or speak it back using an assistant like Cortana on Windows. A con of this approach is that the tablet is mounted to the dash, meaning the user must look down to view the information whereas our implementation allows the user to continue to look at the road or environment in front of them as information is added to the HUD.

2.5 Proposed Design

Our solution makes use of the hardware the client initially provided. In order to make an HMI (Human Machine Interface) system that interacts with human eyesight. The team is using the Tobii Eye Tracker 4C and a combination of normal and infrared cameras for our object detection system. In this case, object detection is provided by a Python/C++ based image analysis library called OpenCV. For data projection, we will use projection film that is compliant with Iowa Department of Transportation standards and laws since that is the area where this project will be demonstrated. Voltage input required to power the module is 5.5 V at minimum and will be through means of an electrical outlet for the time being. The entire system is tied together on the Nvidia Jetson TX2, an ARM based platform.

We currently have the Nvidia Jetson TX2 up and functional with ubuntu. On the Jetson TX2, we also have corner detection working with a webcam and OpenCV. We tried getting a Tobii Eye Tracker 4C to work in this system but ran into issues, namely that we have object code compiled for an x86 system while we are using an ARM system.

This issue is where our different designs come in. One solution is to try to get the Tobii eye tracker working natively on the Jetson TX2. We can do this by either getting an object file from Tobii or via virtualization/emulation of a x86 system. We can go through a microcontroller that has an x86 architecture such as the Intel Quark Microcontroller.

Functional requirements of the module are:

- Can we identify lanes on the road
- Can we identify important objects on the road(signs, people)
- Can we track user's eyes in the vehicle
- Can we project this in a heads up display style
- A system that will track eyesight over a large visual field
- Can determine whether a sensor has been compromised

Non-functional requirements are:

- Object detection and eye tracking to be done simultaneously at real time
- Object detection will to be done with an acceptable level of accuracy
- Fast, reliable and clear HUD capable of keeping up with user while they do the task at hand
- Reliable and secure module protected from malicious users and hackers.
- A user friendly calibration session for each user profile as pupils may differ due to different socioeconomic traits.

In order to achieve these requirements we need to follow the Iowa Windshield obstruction laws.

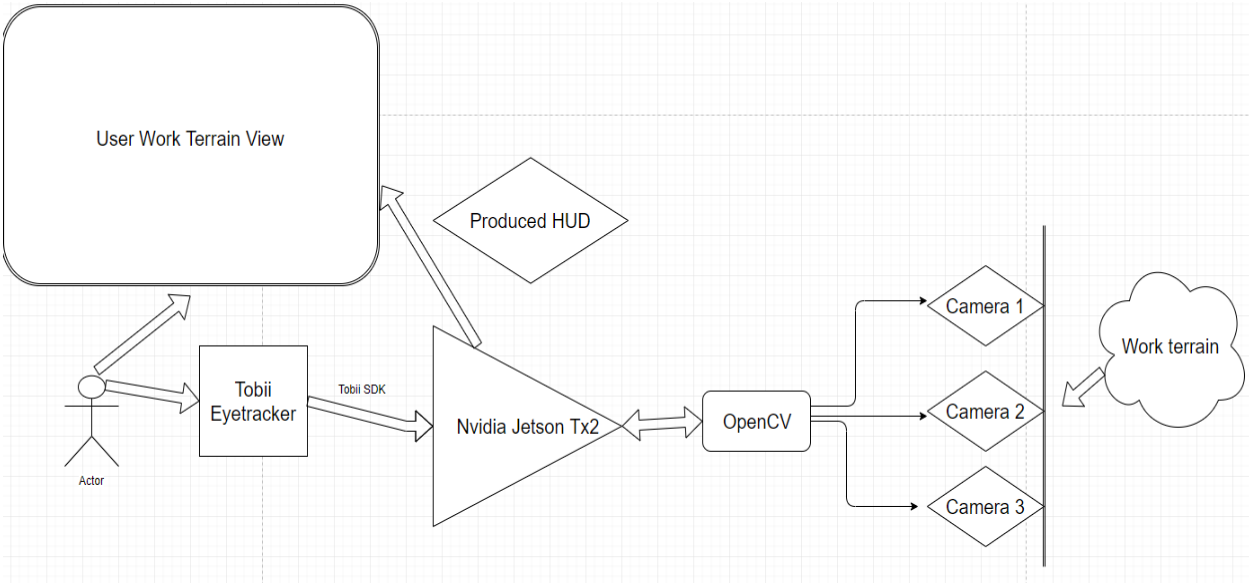

Figure 3: Hardware Design with Eye Tracker Connected to Jetson TX2.

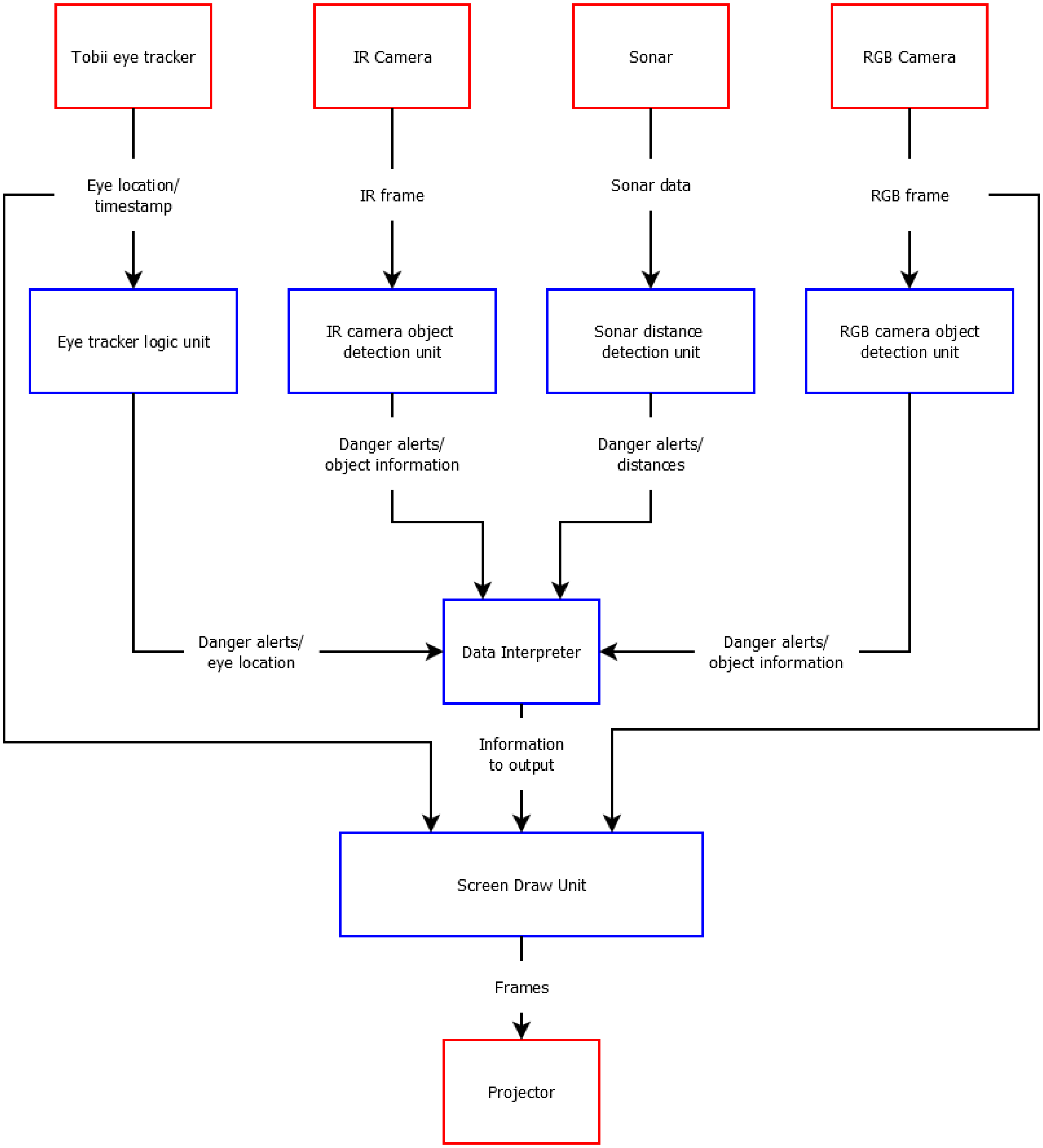

Figure 4: Software Design Diagram.

The analysis of our design starts with the requirements given to us by the client. We are to use an eye tracker to communicate with some module that will queue relevant information to a display. Therefore, we've come up with a potential architecture that obeys the limitations imposed on us as seen in *1.5 Assumptions and Limitations.* Figure 3 shows one of our hardware designs and figure 4 shows the software design.

In the proposed design, the Tobii eye tracker takes input from the users eyes. This information is transferred to the Nvidia Jetson TX2 which is monitoring the work terrain at the same time. It is important to note that the position the user is looking at in the User Work Terrain View is the position the Tobii eye tracker will communicate to the Jetson TX2, as the purpose is to monitor whether the user is keeping track of certain things in the designated User Work Terrain View. The Nvidia Jetson TX2 would be taking input from 3 cameras and parsing the video input frame by frame using OpenCV to detect objects that are relevant to the user's purpose. This imaging is then returned by the Nvidia Jetson TX2 to the User Work Terrain view so it can simulate a Heads up display (HUD) to notify the user of certain things. This defines our flow for the system. However, these arrows merely represent information casted from one device to another. We are not taking into consideration how they communicate. Tobii Eye tracking uses software (TobiiSDK) that only supports x86 architectures, whereas our Nvidia Jetson uses an ARM based processor. Therefore, the drivers that Tobii needs at some of the lowest levels of operation will not be understood by the Nvidia Jetson Tx2 and thus will not be installable there. It will simply not work on the Nvidia Jetson TX2. Since it has not been determined yet whether we can directly connect to the Tobii Eye Tracker to the Jetson, we have decided to utilize different approaches:

1) Introduce a mediary between the eye tracker and the Jetson
2) Use an ROS Wrapper to convert x86 instructions to ARM instructions
3) Swap out the Tobii eye tracker for one that has ARM based processor drivers readily available
4) Use Gaze SDK rather than Tobii SDK

By introducing a mediary, we mean introducing a microcontroller with x86 architecture that can communicate with the Tobii eye tracker in the stead of the Jetson. It will then transfer the information from the users eyes to the Jetson via the conventional UART(Universal Asynchronous Receive & Transmit) device. Figure 5 shows this design.

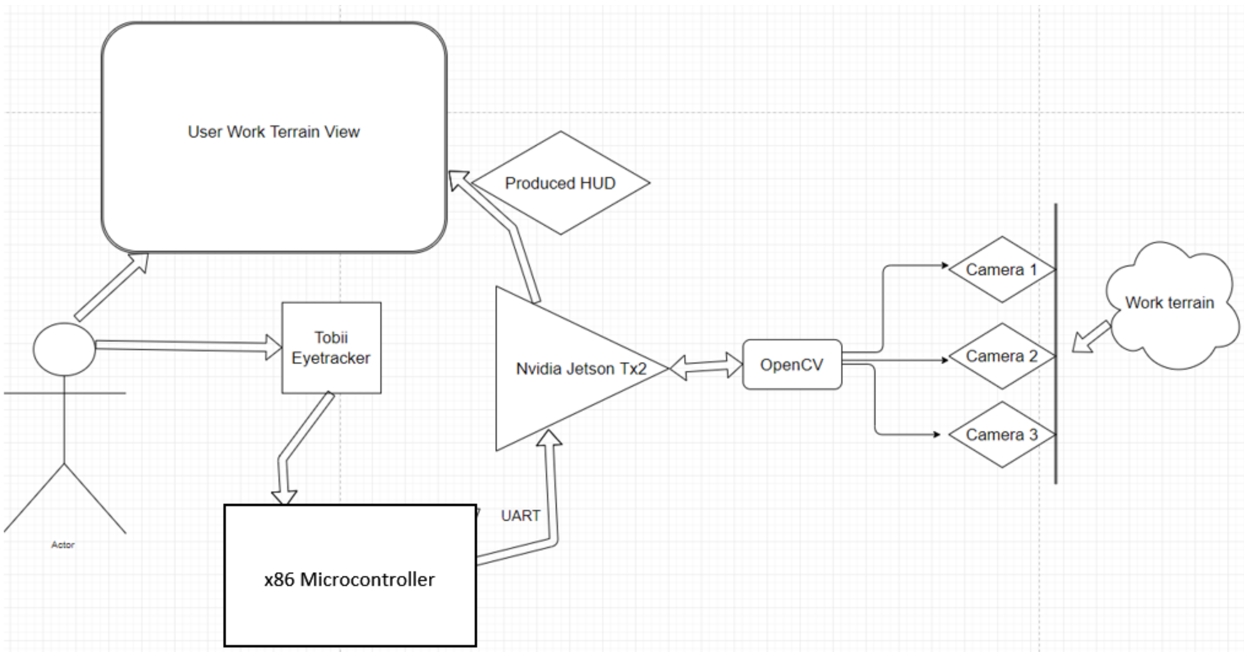

Figure 5: Hardware Design with Mediary Microcontroller.

Later in our research, it has been revealed that we should look into ROS wrappers to drive the eye tracker on ARM. This can be represented as a straight arrow on our diagram as a potential solution alongside using the Tobii Gaze SDK software. Now, we will have to determine what cameras we should use. Various use cases specific to our client may include muddy situations or other agricultural environments that may be in the dark or light. Hence, more than one type of camera must be utilized for the purpose of object detection. So we replace Cameras 1, 2, and 3 with RGB, IR and depth camera respectively. RGB is the regular camera view we are used to. IR camera is to detect edges of objects, and a depth camera may be used to measure the distance. This depth camera may actually be substituted with different sensors such as sonar. These sensors will also have code that will check the validity of the data. If the data is deemed wrong we will indicate the user of the potential issue with the sensor.

Finally, we introduce what we will use to make our HUD. Our HUD will be displayed information depending on their occupation. For example, If they are driving a vehicle then it will be the windshield. How we hope to achieve this is by using a projector to project the information onto transparent projection film that obeys standards set out by the Iowa Department of Transportation. This leaves us with the diagram shown in figure 6. This figure accounts for using either a mediary microcontroller or connecting the eye tracker directly to the Jetson TX2.

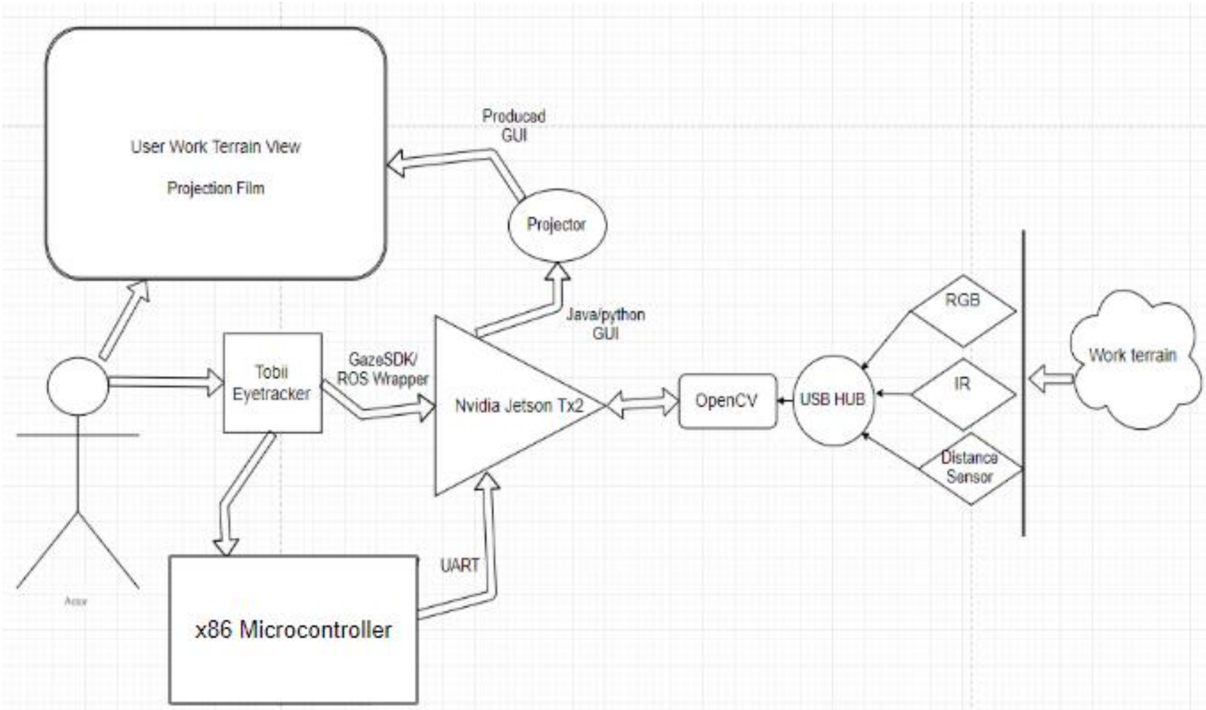

Figure 6: Detailed Project Design.

This prototype is the best way to achieve this purpose. On one hand, there is plenty of hardware where the work needed to be done will be divided. The only multitasking device here would be the Nvidia, as it will be running a kernel with the required minimum dependencies, taking input from the cameras and parsing them frame by frame in search of scripted objects. This is where we believe a bottleneck may arise. The Nvidia Jetson TX2 is rated to be the best for this purpose as it has been utilized for a gaze tracking drone (Yuan 1) . However, for our purposes the Jetson is parsing video input from 2 different channel inputs that are connecting via Universal Serial Bus port. This may cause traffic data and may be a potential weakness in our architecture.

2.6 Technology Considerations

The Jetson TX2 is the fastest, most power-efficient embedded AI computing device. Built around NVidia Pascal, it is loaded with 8Gb of memory and features a variety of standard hardware interfaces making it exceptionally suitable for our project, it be declared that it is not an inexpensive piece of equipment.

The Tobii Eye tracker is a multipurpose eye tracker usually utilized in means of gaming. It features embedded processing and seamless integration through means of a usb port and can interface with most Operating systems.

Both of these pieces of equipment proved to be rather expensive, but they may change depending on the abstraction of this project.

Our team has taken numerous counts of technological devices into consideration, however one that may prove extremely useful can be a depth camera. Although it fits no where in our project as of yet, it may still be used for face recognition that is connected to the eye tracker or to retrieve a certain profile of an object on the other side. In addition, using glasses would ease the process of detecting the users gaze however the client specifically requested that we not use any eyewear.

2.6.1 Technical approach considerations

Our technical approach is mainly to use the given module's Operating system to interface with the Tobii eye tracker. We will use the open source code to provide us with many options and ways to interact with the physical hardware and display the required information to the user. There will be various scenarios that require different forms of attention and so we will utilize whatever peripherals we are provided with to achieve the purpose. As the prototype progresses and grows, we will utilize different testing environments to further manifest any changes needed and deliver the prototype.

There may be necessary technical support and so our team will be in close contact with the different areas of expertise (Such as NVidia, on campus VRAC, Electronics Technology Group), and if permitted we will hopefully use some of their hardware/software debugging methods to overcome any obstacles in our way.

2.6.2 Testing Requirements Considerations

The testing requirements considerations for the project are as follows: the project will have to run on the hardware that is provided. A predetermined data set will be obtained to give testing a consistent stream of data to work with and test expected versus actual resulting display from the program.

There have been several environments where this project maybe considered, however for the sake of this class and time appropriated by the client these scenarios may be left ultimately unexplored.

2.7 Security Considerations

The security considerations for this project are minimal. The project should run independently of an internet connection so the project should be immune from most common attacks through that connection. All physical hardware will be stored in the senior design lab in a locker secured by a lock. This will prevent any theft of materials during the course of the project.

It will convey the maximum threshold of accuracy so that safety is ensured in usage as well. Failure of this device's reliability may result in undefined behaviors that may cause harm.

2.8 Safety Considerations

The state of Iowa requires that any tinting of the windshield must admit at least 70% of the light through. Our project will be using projection film on the windshield of a vehicle. Because of this we must follow the Iowa standard to ensure the safety of the operator of the heavy machinery and the public around the vehicle. Any projection film chosen for the project must conform to this standard as heavy machinery such as tractors and combines are often driven on roads to transport them from one location to another. This can be seen in figure 2 in the assumptions and limitations section.

Additional safety considerations include the testing environment. Rather than testing this on heavy machinery, the team will test using an all-terrain vehicle. This is due to the requirement of special training to operate a combine. This will reduce the risk of injury to others in the testing environment and will allow for more quality testing to be conducted during the project timeline.

2.9 Task Approach

In the proposed design section, Figure 4 and 6 show the basic block design of the hardware and the software components. The hardware side of things is simplistic. All of the sensors need to connect to our microprocessor and then the microprocessor will output the results onto a screen.

The complex part is on the software side of things. We need to implement a computer version component that will interpret what the cameras sees. We then have to link that information together with the other sensors and make decisions upon all of the information.

2.10 Possible Risks and Risk Management

The main risk of this project is the newness of the concept. This is new territory for what is being done with much of the system in many ways is new. This team is also new to all the hardware being implemented which adds to the risk. The major risk is that the project will fail due to over ambitious use cases. This will be minimized by ensuring the client is up to date with all limitations of the software and the hardware with the intent of updating deliverables and/or functional requirements to meet any development in the limitations found.

2.11 Project Proposed Milestones and Evaluation Criteria

Milestone 1: Tobii Eye tracker connected to Jetson TX2. This milestone's evaluation criteria is that the eye tracker is connected to the Jetson TX2 and working.

Milestone 2: Issue detection completion. The criteria for this milestone shall be that an issue is detected by the system and relayed to the user.

Milestone 3: HUD display completion. The criteria for this milestone shall be as follows: the eye tracker correctly tracks the user's eye and the information laid out in the functional requirements is displayed to the user appropriately.

Milestone 4: Component integration. The criteria for this milestone is connecting all of the systems together to build a single complete system.

2.12 Project Tracking Procedures

This project will use Trello to track the status throughout the lifetime of this application. The Trello board will have different categories to track work as the project progresses. These categories are project backlog, in development, testing, finished, and a list of known bugs within the project. This organization will allow for easy tracking of who is working on what as team members can be assigned to various tasks on the Trello board.

Direct communication regarding the project will happen using Slack. There will be different channels that correlate to the different fields we have defined in Trello, this will give the members of this team a more direct form of collaboration and will ease their interface as they work towards completing the purpose of that field.

2.13 Expected Results and Validation

The expected result is complete efficiency in tracking the eye and delivering information based on its position. The way we will validate this execution is by passing it to a volunteer who would try and utilize it in an appropriate workspace. Our confirmation would be their performance before and after utilizing this eye tracker and the margin of that difference will determine its effectiveness.

2.14 Test Plan

To test our system, unit tests will be used. These will allow for a swath of standardized tests to be implemented during the course of production to ensure that all aspects of the project work together and adhere to the method, including functional and non-functional requirements. One way to test the project is using a predetermined video input. The product will be tested using a sample loop of video with objects and input data such as speed, tire pressure, etc. that will be consistent throughout the production of this project. The video shall be taken from the vantage point that the real-world usage will take place. It will include road signs, obstacles, and lanes on the road. This will allow for a baseline to test against as the team will establish which object types shall be detected in the video and will be able to match if the application detects these. Additionally, it will enable us to display the information from the vehicle onto the windshield. Because there will be a prerecorded video, we can test cases where the driver does not look at the alert, looks away for too long, or a range of activities that are not safe for the driver to be engaging in while driving heavy machinery.

Below is the flow diagram for the project. A feature of the project will be produced from the product backlog and marked as needing testing. The feature will be subjected to a series of unit tests to ensure that the code performs as it should. Then it will be evaluated by the team to ensure that all functional and non-functional requirements have been satisfied. If it has the feature will be marked as completed. If not, identify the requirements not met. The team will brainstorm ways to satisfy these requirements and the feature will be updated and bugs fixed. This loops until such a time that the requirements are met, and the unit tests are passed.

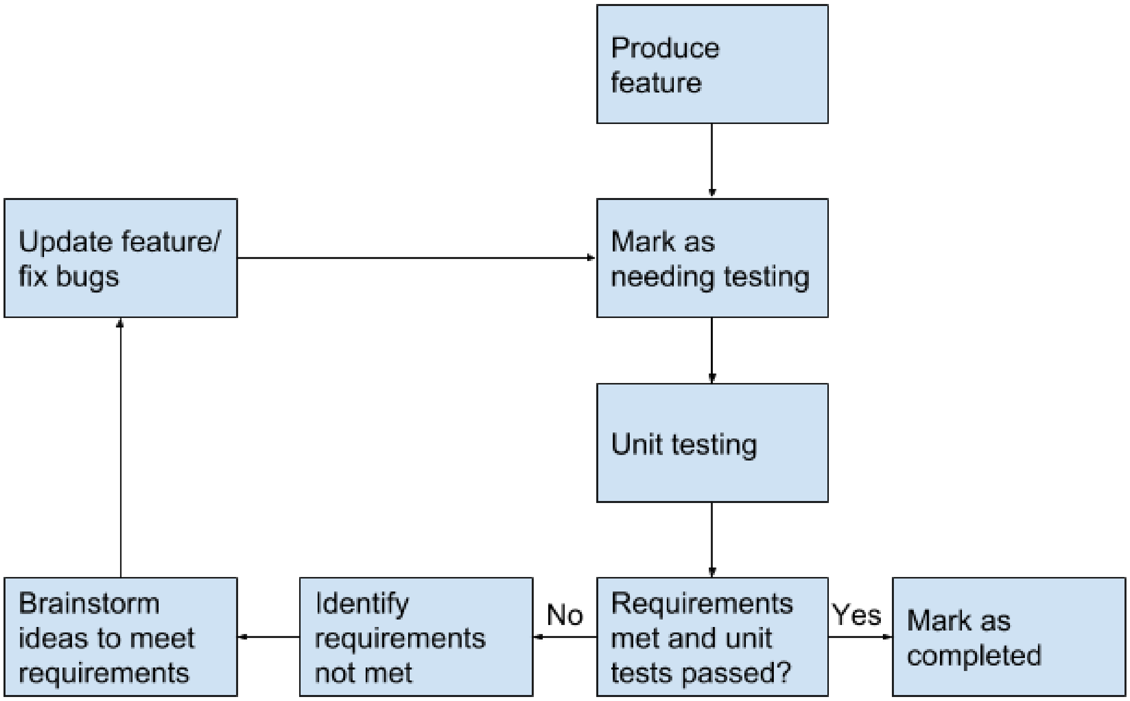

Figure 7: Flow Diagram.

3. Project Timeline, Estimated Resources, and Challenges

3.1 Project Timeline

In the first semester our primary task will be researching and testing the equipment we have. We will spend the time looking into the software required to control our eye tracker and cameras. Afterwards we will start testing the equipment and validating them for our purposes. In April we will start working on getting the eye tracker to the location of the user's eyes. Since we plan on using an already developed system, this shouldn't take very long. Afterwards we will start working on getting the object detection system working. This system will be designed by using a normal camera and an infrared camera.

In the second semester we will start by finishing up the object detection system. The next task we need to accomplish is the issue detection system. The issue detection system needs to gather all of the sensor data and then determine if something dangerous is happening. It then indicates to the user of any issues. The final large scale system is the one for drawing the HUD. The two subtasks are creating a basic design and then a calculation system that will determine where to draw the image onto the screen.

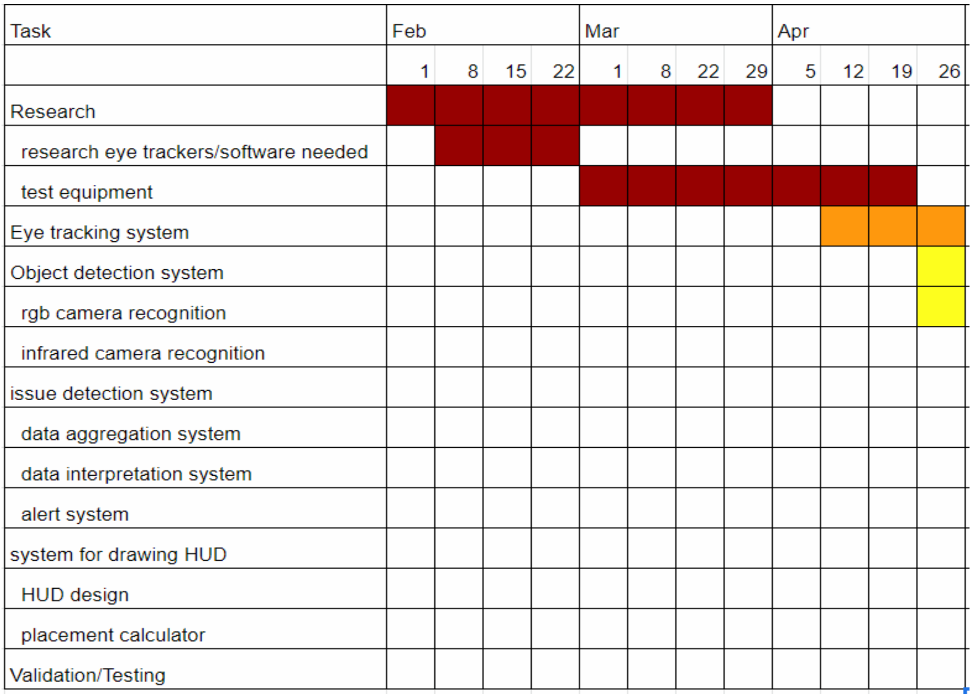

Figure 8: Gantt Chart Spring 2019.

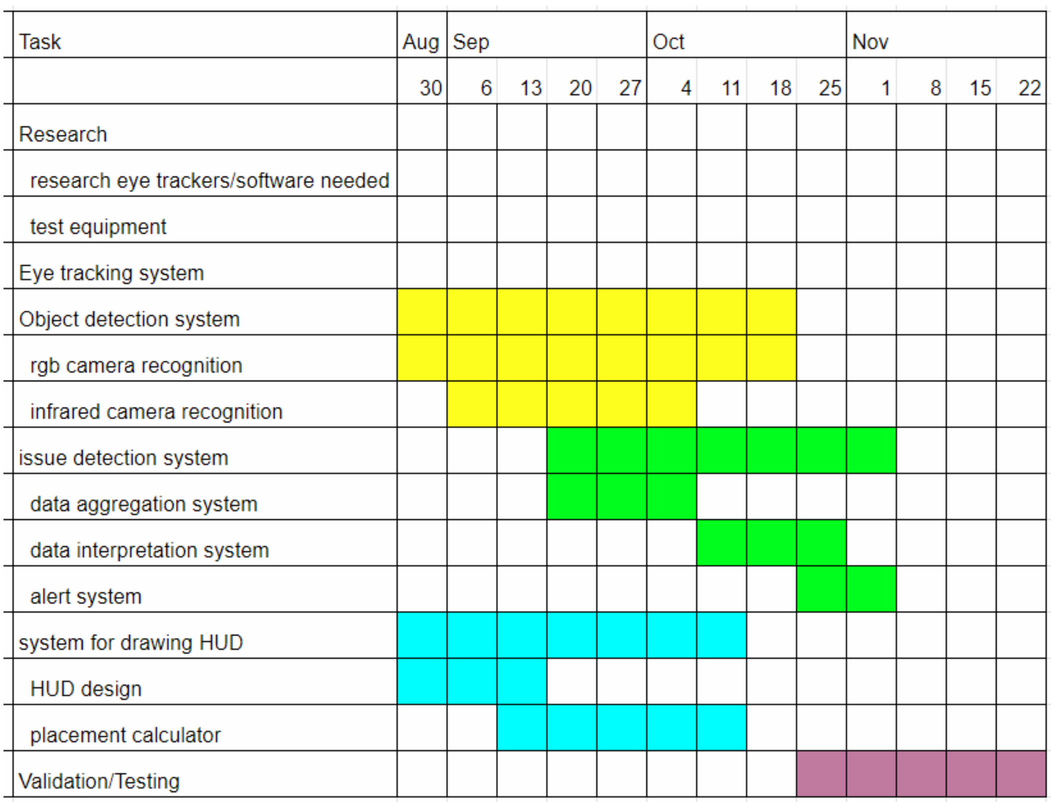

Figure 9: Gantt Chart Fall 2019.

3.2 Feasibility Assessment

There are a lot of separate tasks that need to be accomplished for the project to be completed. This is both a positive and a negative. The positive is that we can test and validate systems individually without too much hassle. The negative is that at the end everything needs to be working properly for the entire system to work. If we get backed up on any system, it could unexpected delays to appear at the end.

The two main tasks that might cause delay is the projector calculations and the lane detection system. The projector calculations will be very tricky since we have to merge the camera information and the eye tracker to figure out where we want to place an image. The object detection system will be the system that will take the longest to get completed. The reason for this is because of how many different things there are in the world. Luckily, we don't need to discern everything so the task should be similar.

Looking at these issues we have determined that a basic test case should be accomplishable in the next 2 semesters.

3.3 Personal effort requirements

Task	Effort (hours)	Explanation
Research	70	Researching and testing the equipment we have.
research eye trackers/software needed	20	Researching will probably be continued throughout the project but most of the decisions should be made at the beginning so we can start designing.
test equipment	50	Testing equipment carries an unknown aspect to it. While we could finish the task fairly fast this has the most possible issues that could pop up. We might need to go back into researching other components.
Eye tracking system	25	The eye tracking system we have is a an already developed system so most of the work should be getting if setup.
Object detection system	105	Object detection is a very complex issue and will probably take the longs amount of time. Thankfully we can vastly simplify the issue by have a set test case.
rgb camera recognition	65	The normal camera will be doing the brunt of the object detection so this would take the

		longest to integrate.
infrared camera recognition	40	The infrared camera is mostly used to supplement the normal camera so it should take less time.
issue detection system	70	The issue detection system is made up primarily of the interpretation system. Given we have a set scenario we only have so much to look out for.
data aggregation system	25	It could be difficult to set up given all of the data that needs to go into the system. It is subject to delays in the other systems.
data interpretation system	25	Interpreting the data could be difficult give the amount of data.
alert system	20	The alert system is mostly setting up connections to the HUD drawing system and the data interpretation system.
system for drawing HUD	20	The HUD could talk a lot of time given our lack of expertise in how to do these types of calculations.
HUD design	20	The general design of the HUD shouldn't take too much time. It's requirements will probably need to be continually added onto throughout the project.
placement calculator	30	The placement calculator could be very difficult to get working properly since we might not be able to test on the system it's design for.
Validation/Testing	40	Testing will be done throughout the project but we need to leave a good amount of time at the end so we can both find and fix potential issues.

Table 1: Personal Effort Requirements.

3.4 Other resource requirements

The resources needed for the project are provided by the client. These resources are a Nvidia Jetson TX2, Tobii eye tracker, front projection glass, projector, and camera. Any additional resource requirement that are discovered will be provided to the team by the client as appropriate to the advancement of the project.

Peripheral resources may also come in request as for different environments, this application may tailor differently in different environments.

3.5 Financial requirements

At this time no financial requirements have been identified beyond what has been provided to the team by the client. However, the end product should be as least costly as possible as this solution may be duplicated into multiple departments or users. Proper design and architecture will be required to mitigate costs and effort.

Cost of Device (approximately):

TX2: $540

Tobii Eye tracker: $150

Generic Projector: $80

Projection Film: ~$40

Cameras: ~$300

Intel Compute Stick: $120

Total Cost: $1,230

4. Closure Materials

4.1 Conclusion

Today's world is becoming more and more integrated with information. One area where this information could be most useful is while operating large machinery that allows the user to see all the relevant information at a glance without taking their eyes off what is right in front of them. The goal presented throughout this document is to create a heads up display that utilizes eye movement to intuitively control information presented to users.

The approach laid out about shows the main aspects needed to implement the solution are an eye tracker with the Jetson TX2. Additional features will be added through camera support. The key features of the project are well defined and will ensure that the final project is feasible and possible within the timeline presented in this document.

4.2 References

1. Duchowski, Andrew Ted. *Eye Tracking Methodology: Theory and Practice*. Springer, 2017.
2. "Embedded Solutions for Drones & UAVs from NVIDIA Jetson." *NVIDIA*, www.nvidia.com/en-us/autonomous-machines/uavs-drones-technology/.
3. "Automotive." *Tobii,* 2019, www.tobii.com/tech/products/automotive/.
4. "Jetson TX2 Module." *NVIDIA Developer*, 5 Dec. 2018, developer.nvidia.com/embedded/buy/jetson-tx2.

5. “UXLab Inventory.” *Virtual Reality Applications Center*, 11 Apr. 2016, www.vrac.iastate.edu/facilities/uxlab/uxlab-inventory/.

6. “Window Tinting Standards.” *Iowa Department of Transportation,* https://iowadot.gov/mvd/resources/windowtintingstandards.pdf.

7. “Gaze: the driver-monitoring eye-tracker software which could be in your next car.” *Auto Express,* https://www.autoexpress.co.uk/car-news/103614/gaze-the-driver-monitoring-eye-tracking-software-which-could-be-in-your-next-car