

Secure Testing Headset

Design Document

Team: sdmay21-01

Client: Cornerstone Strategies, LLC

Adviser: Dr. Rover

Team Members (Roles):

Rob Barton (Facilitator), Morgan Ambourn (Scribe),
Asa Pauls (Hardware Lead), Ehren Fox (Software Lead),
Nathan Andersen (Report Manager), Zach Johnson (Test Engineer)

Team Email: sdmay21-01@iastate.edu

Team Website: <https://sdmay21-01.sd.ece.iastate.edu>

Executive Summary

Development Standards & Practices Used

Materials used for the housing of the product will either conform to California proposition 65 regarding levels of toxic chemicals present or will come with a warning regarding their use.

Our power supply will meet the US power plug standards and work in a standard NEMA 5-15 outlet.

We will be using C code with adhering to standard style guidelines as defined at https://www.cs.swarthmore.edu/~newhall/unixhelp/c_codestyle.html.

Our software platform will comply with Question and Test Interoperability (QTI) standards that ensure that test content is formatted correctly and can be transported between testing platforms and authors of the content.

The design of the project complies with IEEE International Standard - Systems and software engineering -- Software life cycle processes. This purpose of this standard is to define the stages of a software project (supply, development, acquisition, operation, maintenance, and disposal). It ensures a smooth transition of our project from our development phase to future groups working on the project.

Summary of Requirements

- Headset with user interaction
- Securely receives and transmits data
- Can be used to take an exam

Applicable Courses from Iowa State University Curriculum

- Computer Science 309 - provided team work experience and big project experience
- Computer Science 319 - provided experience in UI development via HTML and JavaScript
- Computer Engineering 288 - provided familiarity with working with data sheets provided by a vendor that enabled us to correctly port I/O
- Computer Science 252 - taught shell scripting and basics of linux file setup
- Computer Engineering 308 - more linux basics, navigation, remote connection

New Skills/Knowledge Acquired Outside of Courses

Knowledge

- Cellular vs wifi speed and US coverage
- Optics (how lenses work, corrective lenses, binocular vision)

Skills

- How to connect/use LCDs and Arduino
- 3D printing
- Existing testing software
- NodeJs server
- Running terminal commands on bootup

Table of Contents

1 Introduction 7

- 1.1 Acknowledgement 7
- 1.2 Problem and Project Statement 7
- 1.3 Operational Environment 8
- 1.4 Requirements 8
- 1.5 Intended Users and Uses 9
- 1.6 Assumptions and Limitations 9
- 1.7 Expected End Product and Deliverables 10

2 Project Plan 11

- 2.1 Task Decomposition 11
- 2.2 Risks And Risk Management/Mitigation 12
- 2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria 12
- 2.4 Project Timeline/Schedule 13
- 2.5 Project Tracking Procedures 14
- 2.6 Personnel Effort Requirements 14
- 2.7 Other Resource Requirements 15
- 2.8 Financial Requirements 15

3 Design 16

- 3.1 Previous Work And Literature 16
- 3.2 Design Thinking 18
- 3.3 Proposed Design 20
- 3.4 Technology Considerations 23

3.5 Design Analysis 23

3.6 Development Process 24

3.7 Design Plan 24

4 Testing 27

4.1 Unit Testing 27

4.2 Interface Testing 28

4.3 Acceptance Testing 30

4.4 Results 31

5 Implementation 32

6 Closing Material 33

6.1 Conclusion 33

6.2 References 34

6.3 Appendices 35

List of figures/tables/symbols/definitions

- Page 13 - Figure 2.4.1, Gantt chart depicting project timeline
- Page 15 - Figure 2.6.1, Personal Effort Requirements
- Page 16 - Figure 3.1.1, Google Cardboard
- Page 17 - Figure 3.1.2, Project North Star Headset Design Rendering
- Page 19 - Figure 3.2.1, Security Brainstorming
- Page 19 - Figure 3.2.2, Secure Testing Brainstorming
- Page 22 - Figure 3.3.1, Initial Sketch of Headset
- Page 26 - Figure 3.7.1, Hardware and software diagrams
- Page 35 - Figure 6.3.1, Off/on button for headset
- Page 35 - Figure 6.3.2, UI for entering network name
- Page 36 - Figure 6.3.3, UI for entering network password
- Page 36 - Figure 6.3.4, Tao platform login page
- Page 40 - Figure 6.3.5, Initial Headset
- Page 40 - Figure 6.3.6, Open case with screws
- Page 41 - Figure 6.3.7, Open showing frame with screens
- Page 41 - Figure 6.3.8, Exposed GPIO Headers on Pi
- Page 41 - Figure 6.3.9, USB Extender with Dongle
- Page 41 - Figure 6.3.10, USB Extender with Trackpad
- Page 42 - Figure 6.3.11, Example of Tucking Excess Cabling
- Page 42 - Figure 6.3.12, Returning Screen Frame

1 Introduction

1.1 Acknowledgement

We would like to thank Rachel Schoenig, from Cornerstone Strategies, and Dr. Diane Rover, from Iowa State University.

1.2 Problem and Project Statement

Cheating of test takers may be increasing with increasing use of online exam environments. Tests are required to confirm that people are qualified. Educators and testing centers want to ensure that if a user passes a test, they did so because they learned and understood the content being tested, not because of outside resources.

Some tests can be with high stakes, which can create an incentive for people to cheat, namely for personal or financial gain. Stolen answers sold to other test takers compromises an exam's merit. Tests can cost a lot of money to develop, so testing industries want to ensure that questions are not stolen & distributed.

We've experienced the increase of difficulty moderating test takers as Covid-19 has forced educators and testing centers to move to online platforms instead of in person testing. One issue common with online proctoring is that there is a disconnect between the test, the camera, and the student's face. Cameras cannot see the full field of view of the user. Even for in person proctored settings, high-density testing areas make wandering eyes easy to hide.

An additional limitation of existing testing technologies is the ability to bypass a lockdown browser with a virtual machine (vm). With a vm running in the background, test information can be captured and transmitted. Ensuring the identity of the user taking the test is also an existing challenge.

We're aiming to improve upon current online testing tools that simply monitor students with the use of a webcam and make use of lockdown browsers. Our design is a wearable headset capable of delivering the test with neither the user interacting with the outside environment nor others interfering with the testing environment. We are aware that our product will not be as necessary if open book/note exams become more popular, but we will still have a market in the standardized test world.

1.3 Operational Environment

We expect our product to typically be used indoors. It may be dropped or knocked off a table. We may want to consider the possibility of a glass of water being spilled on or near our device, but this isn't necessary. Refer to the risk-mitigation section for further details on potential operational risks.

1.4 Requirements

Functional Requirements

The product will allow the user to power on and off the headset. The user will be able to connect to the internet and navigate to the testing driver for their specific exam. The exam will be displayed and can be interacted with. The exam can be scrolled through and answers can be selected. When the user is ready, they can proceed to the next question. At the end of the exam, the user will be able to submit their exam, and begin another exam if necessary. The product will limit the user's access to only the test. The test will only be viewable to the user, which means there will be no outside access to or extraction of test content. Only the user will be able to interact with the test. Content between the input device and the headset will be secure. Any physical tampering with the headset will be evident.

Non Functional Requirements

The headset must fit on the user's head with the ability to adjust the size. The headset cannot overheat during use. The headset must remain charged for the duration of use. Previous internet connections must be remembered for user ease. Multiple testing drivers must be accessible. Pages must load within 3 seconds. The headset must be durable and water resistant to sustain general use. The headset must be reasonably priced so that it can be purchased by a student. The user interface must be intuitive without need for specific instructions on how to navigate.

Standards

The relevant standards for this project include Question & Test Interoperability (QTI), IEEE International Standard - Systems and software engineering -- Software life cycle processes, and C Code Style Guidelines - Swarthmore Computer Science Department.

QTI standards ensure that test formatting works across platforms and systems.

The software life cycle processes standard defines the stages of a software project. These stages include supply, development, acquisition, operation, maintenance, and disposal. Our project is currently in the development stage. Following our work on this project, the product will be transferred to another group for further development and production.

The C Code Style Guidelines outline the use of error handling, clear naming conventions, defining constants whenever possible, providing comments, and writing modular code.

Constraints

The constraints of our project included cost, time, power, and weight.

The cost of the prototype had to be under \$1000, which limited our choice of raspberry pi that we could use. To be conservative with price to ensure room extra expenses, we chose the cheapest raspberry pi that would still provide the required features. The price of the final product is about \$190 for the prototype and an estimated \$120 for the production version. Neither of these costs include packaging, technical support, or customer support. The estimates for future product cost are based on an average 17% off for items bought in bulk from various suppliers.

The time constraint required a functional prototype within a year. This time constraint influenced the end features of this model. One decision was to remove sensors and detection of cheating from the end prototype this iteration.

Weight was a constraint based on user comfort studies. The suggested weight is under 400g. The actual weight of our headset came to about 393.55g, but would have had an additional 60g if the touchpad input source was also added to the headset.

Power was a constraint because the product must be functional at any point in time. Continuous charging was determined as the power source. Because weight was also a constraint, a battery within the headset was not feasible as it would put the weight above the requirement.

1.5 Intended Users and Uses

We intend our product to be used by students and those seeking certain certifications. We hope this device is employed by college entrance exams such as those offered by the college board, internal college exams, and certification exams such as the BAR and CPA exams.

1.6 Assumptions and Limitations

Assumptions:

We can assume, for the sake of prototyping, our product will be used exclusively by English speakers in the U.S. The device will be used from remote locations i.e., not necessarily in a testing center. The device will be communicating with an internet server, and that the user will have wifi access.

Limitations:

Our prototype must cost under \$1000 to make unless we obtain resources from other sources such as ETG. Our device being a wearable headset presents several limitations pertaining to user comfort such as weight, heat dissipation, and vision impaired user constraints. The

maintenance of a secure connection limits how we can connect to the server. Our construction should be tamper resistant and tamper evident. Finally, we are limited in our ability to accept input since we cannot allow external devices to contact the device. External devices such as a keyboard could exit lockdown browser mode, which compromises the security of the test.

1.7 Expected End Product and Deliverables

The end product will be a device that allows for an exam to be displayed on screens. The screens will be minimally sized to reduce total headset weight, but must still be readable. The device shall have input in the form of a hand held controller that will be able to select answers in a multiple choice exam. The headset will be enclosed to ensure no access to outside resources. The device will not be powered by a battery for minimum viable product. It will instead be directly connected to power for the entirety of the device's use. The device shall be durable enough to survive minor damage and should be tamper evident so that testing security can be determined at a later date. Delivery date for the device shall be no later than April 27, 2021.

A document shall be provided that explains how to use the device, how the device was developed, what future iteration features would include, what the development and production expenses would be, and what research went into the development of the product. The document shall also include guidelines for edge cases for the device such as how the device handles a disconnect from its power supply or a network connection.

2 Project Plan

2.1 Task Decomposition

For hardware we have a few main tasks. We need to have input so that users can interact with the display. The device has to have the ability to connect to a wifi network to be able to communicate and exchange information from the testing server or service. The device needs a display that is able to display the test, and update based on the user's interaction from the inputs. Finally the device will need to have a power system that is able to power every component of the device.

In addition, there is some software needed. We need to make sure that the testing information is delivered securely to the device, and that the answers are transferred off the device securely. We also need to allow the user to input wifi network information without giving them access to the terminal or desktop. The user interface should be simple, but understandable so that the user isn't distracted while taking their exam.

Tasks and subtasks, as well as interdependence among tasks, are listed below:

- Hardware
 - Hand held controller
 - Wifi connection
 - Data transfer
 - Secure data transfer
 - Power
 - Screen/optics
 - Display an image
 - Update image based on user interaction (ensure high enough frames/second)
- Software
 - Receive data from a server and display
 - Quality user interface for exam
 - Accommodate limited user input
 - Enable network connection

Each of the above tasks has some dependence on the other tasks as described here. The hardware to be used must be obtained before input can be implemented or data can be transferred to and from the device. The screens and optics must be established before headset enclosure is finished. The list of items above is arranged in a way that shows loose dependencies between tasks. Certain steps can also be developed in parallel. For example,

software can be worked on before the hardware is complete since there will be enough of an environment for testing during development.

2.2 Risks And Risk Management/Mitigation

We outlined a few risks at the beginning of the project that we might experience while designing and developing this product. The project's scope could be too great for the given time frame, if that was the case, then we had planned on narrowing down our scope as necessary. If the project wasn't on pace with our planned schedule, we could make use of the buffer time we built into our schedule and alert our client to the possibility of running out of time. If the cost of development became too great, we planned to limit the features or reach out to the Senior Design program for resources. If our team came across something that required knowledge that none of us possessed, we had planned to reach out to ISU professors for help and guidance. We planned to implement outside security software in case we are unable to secure the device adequately. We did end up running out of time and weren't able to deliver all of the functionality we had planned, and one thing not considered in the original planning was issues when integrating separate tasks such as software and hardware. However most of the risks we faced were mitigated. We reached out to faculty with questions and prepared recommendations for what professional software could be pursued after this class to make up for our falling short.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

These are our proposed milestones:

1. Power supply
2. Displays
3. Split into software and hardware to work in parallel
 - a. Collect network information
 - b. User input hardware
4. User friendly interface
5. Secure connection
6. Optics

Our first milestone is to get the device working which involves getting it plugged in and powered. Then we need to install the screens so the device is capable of output. Once those steps are finished our team can begin developing a method to collect network data and a method for the user to input information into the device. These two tasks can work alongside each other. Once complete we'll need a proper user interface that can still collect and store information while accommodating simplified inputs and maintaining security. With the interface down we should

make sure the device can communicate with the test server securely. And finally if we have time we'll look into supporting a wider range of eyesight capabilities.

2.4 Project Timeline/Schedule

The first semester was spent on pre-project planning. A week went to defining the problem and another to brainstorming solutions and the rest of the semester was devoted to narrowing down what exactly we would attempt to build and how it would work. As part of our agile inspired work cycle looking forward we then broke down our prototype construction into 1-2 weeks blocks that end off with the integration of a new feature. These blocks in order of expected start, are Designing a UI to be displayed by the pi, building a power system to run the pi, attaching screens so the Pi can output, Developing controls for user input, and design a secure data transfer system. After our prototype is built we hope to test it and then reiterate over the design. The Gantt chart below lays out when we plan to do all of this and how long everything is expected to take.

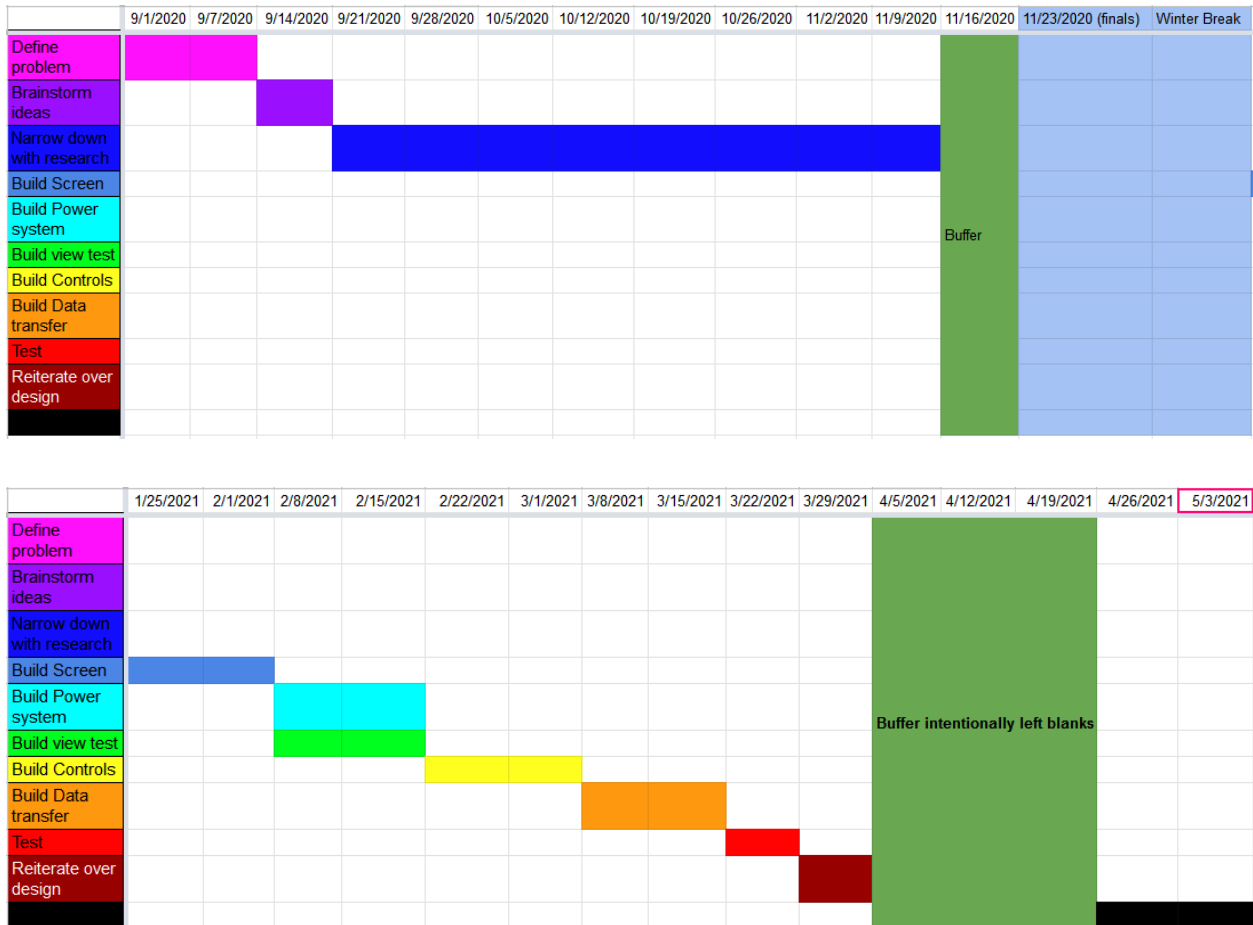


Figure 2.4.1: Gantt chart depicting project timeline

2.5 Project Tracking Procedures

We plan on using Gitlab as our code repository and to track software issues. We will be using Webex Teams and email to communicate between team members, the industry connection, and the faculty advisor. We plan to work in an agile environment. The metrics we will use to track progress will be whether or not the tasks are done as well as to what extent the tasks pass all unit tests.

2.6 Personnel Effort Requirements

We have various tasks split up into reasonable milestones. Some tasks are slated for before the development of the product such as defining the problem and brainstorming ideas. Figure 2.6.1 shows all of the tasks and their expected time requirement. For defining the problem, we expect it to take roughly 8 person-hours. We expect to have a meeting with our industry contact, go over the idea, and together define the problem we are trying to solve. Brainstorming ideas is the next step before our development, we expect it to take roughly 6 person-hours, enough for the team to come together and pitch ideas all together.

The following tasks on the table, shown in Figure 2.6.1, are all for the development of the product. These times are estimations on the number of person-hours it will take to complete each portion. We expect the screen and the software to take 40 person-hours each, as these two tasks will be a large portion of the project. Working on Data Transfer, being able to transfer data from a server to the device securely, we expect will take 35 person-hours. Working on the UI, taking the data transferred from the server and displaying on the screen, we expect will take 30 person-hours. Implementing controls on the device we expect to take 20 person-hours. We did not devote time specifically to power design as the more involved stand alone portion was scrapped.

Task	Estimated Person-Hours	Actual Person-Hours
Define problem	8	8
Brainstorm ideas	6	6
Build design: Screen	40	14
Build design: UI	30	32
Build design: Software	40	37
Build design: Data Transfer	35	0
Build design: Controls	20	30
Build design: Casing	0	24

Figure 2.6.1: The above chart determines how many hours are expected to complete each task.

2.7 Resource Requirements

This project will require us to procure a 3D printed casing, A Raspberry Pi zero w with accompanying power cord, a separate power switch, lenses, straps, and any input devices we intend to comprise our prototype directly.

To aid in our development we will also need a second Raspberry Pi zero w to allow for simultaneous development on the hardware as well as an online server with content for demonstration and testing purposes.

2.8 Financial Requirements

The only monetary resource available is \$1000 from the client (Cornerstone Strategies). Any additional money needed would have to be requested from senior design funding.

3 Design

3.1 Previous Work And Literature

A similar product to our first semester design is the Google Cardboard, as seen in Figure 3.1.1. According to Google, their product allows you to “experience virtual reality in a simple, fun, and affordable way.” In essence, it is a cardboard box with some lenses that has a slot for you to insert your phone. When running their cardboard app, and holding the device to your face, you get a VR headset device.



Figure 3.1.1: Google Cardboard device with a phone.

Benefits:

The Google Cardboard is very similar to what we want to create. Lense setup of the Google Cardboard is something that we could look at in our design. The device is very cheap, which suggests that we can implement a relatively cheap prototype. During development. We can test software on this existing platform.

Drawbacks:

We don't need the extra overhead of VR capabilities, which is a component of the Google Cardboard. Another downside of the device is that it requires users to have a compatible phone to put in the headset. Loading the test onto the user's phone through the Google Cardboard app might not be secure. The user would also have the ability to take out the phone, allowing others to view the test. The user could also record the phone screen to share the test to others, which would be a security risk.

Project North Star is another relevant project, pictured in figure 3.1.2. It is an open source AR project that kicked off in 2018, but has since been discontinued. The goal of the project was to create an affordable VR headset that could be mass manufactured for about \$200.

Benefits:

Project North Star is a useful reference because it has a similar headset housing to what we plan on implementing. Because the project is open source, there are many other related projects available for further research. The original design estimated mass manufacturing of the devices to be about \$200, which would be valuable to reference as we attempt to minimize the cost of our product. Problems that would be encountered during development are likely ones that someone in the project community has already encountered, which would help us minimize roadblocks in our development process.

Drawbacks:

Our project design scope has shifted away from AR and VR. Instead it is simply a head mounted display (HMD). Project North Star is an AR device, which deems much of its design to be irrelevant to our project. We don't need the extra overhead of AR/VR capabilities. An additional disadvantage to this product is that it has been discontinued.

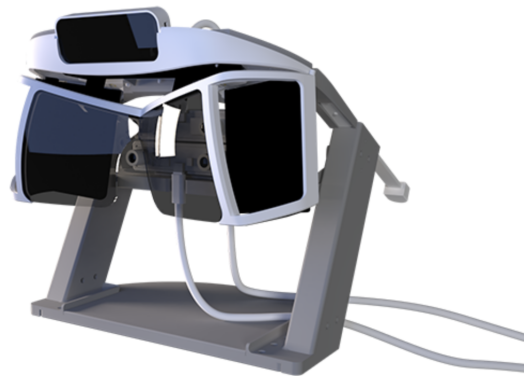


Figure 3.1.2: Project North Star Headset Design Rendering

3.2 Design Thinking

Through the “define” phase of our design thinking, the user will be a person taking a multiple choice exam. Specific types of test takers may include students in higher education or applying to higher education through exams such as the ACT or SAT. The user may also be taking a test for a certification exam. The test they are taking will be multiple choice. To give the users the best experience they should be able to be comfortable while taking the exam, this comfort may not jeopardize the integrity of exam material. We also want to ensure that the test taker is able to interact easily with the test to minimize any stress related to the testing environment; the test taker’s focus should solely be on the exam content. We determined that our product is needed due to current testing circumstances during the pandemic as well as the constantly increasing use of technology for testing.

Figure 3.2.1 and figure 3.2.2 depict the ideation that took place while developing our product. Based on the figures, we narrowed down our product in the following ways. Within our security ideation, we determined that we will not be building custom architecture/OS. We will be using encryption from pre-existing resources/libraries. We will first develop the product to work over wifi, but may later add support for cellular. For our secure testing ideation, we determined that we will not be buying a pre-existing headset due to financial limitations. We will be creating a headset that is not immersive in the sense that the screen content will not adapt based on the user moving their head. The screen will appear as a still image of text, but can be scrolled through. We will not be using gaze tracking or any game controllers, the current hardware we are using is unable to support more device load at this point. The user will interact with the device using a wireless trackball mouse or a trackpad mounted on the side of the headset..

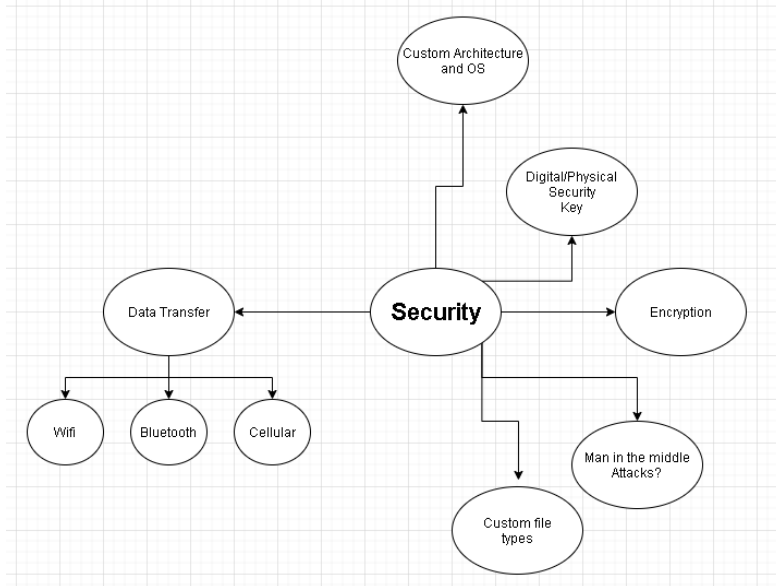


Figure 3.2.1: Ideation about security

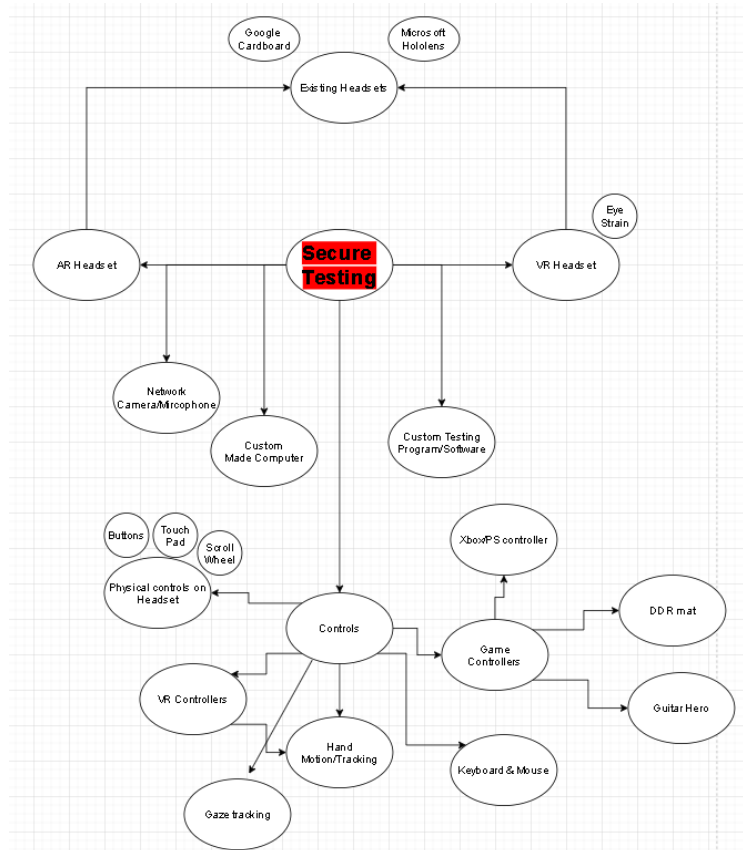


Figure 3.2.2: Ideation about secure testing

3.3 Initial Project Design

Figure 3.3.1 shows the expected look of the headset. The casing will be 3D printed for the prototype, but the final product when produced on a larger scale will be made from a cast mold. The reason for the custom headset is to produce it cheaper than an existing headset that comes with additional functionality that is not needed for testing purposes. The front of the headset around the eyes will have all of the hardware, instead of having some of the equipment on the back of the head, which would require having cords to connect the front and the back. The buttons and scroll wheel will be on the side of the headset. We decided on buttons directly on the headset instead of having extra handheld devices. An extra device could be at risk for being lost/disconnected from the headset. There will be an adjustable strap around the head to ensure the headset remains firmly on the user's head. Within the headset will be two screens, one for each eye. Each screen will be a duplicate of the other. To eliminate the risk of having some type of video recording or image capture inside the headset, there will not be excess room for anything between the user's eyes and the inside of the headset casing. This means that user's will not be able to wear glasses within the headset. To accommodate people with visual impairments or people who usually use glasses, there will be lenses between the user's eyes and the two screens. The lenses will be manually adjustable so that the user can change the focus. The lenses will be adjusted by moving them closer to or farther away from the screen, as well as closer or farther together to accommodate differing pupillary distance.

We are planning on using a Raspberry Pi for the hardware as there are some Raspberry Pi products that come with wifi capabilities. The Raspberry Pi we plan to use is called the Raspberry Pi Zero W, which includes a wifi chip that allows the headset to connect to the internet.

We will be building software that receives the test data from a secure server. The data transfer must be secure, which will be accomplished through pre-existing encryption libraries. Our drivers will need to comply with Question & Test Interoperability standards. QTI describes the standards for ensuring assessments have consistent formats, protocols, and interoperability. The test will be downloaded by the user ahead of time to ensure no dependency on the network bandwidth. The user will select the test they plan to take when ready and can scroll through the question and answer options. The user will select an option using the buttons and can then progress to the next question. The user will also be notified when the battery for the headset is running low.

3.3.1 Revised Project Design

We have kept all hardware in the front of the headset as we had planned for in the first semester. This was for a few reasons. In addition to the reasons we had in the first semester, another big reason was in testing the weight of all the components are currently almost unnoticeable. We did originally plan for using buttons and a scroll wheel as our input device, we changed the input to use a wireless trackball as the sole input device. This change was made for a few reasons, first it is harder to tell what is going on in the headset as no cameras can track what buttons are pushed. The second reason is that we can randomize the starting position of the cursor every so often so even if the trackball motion is recorded the video has no useful information as there is no frame of reference. The potential security risk in the bluetooth or wireless connection was deemed an acceptable risk when considering recorded input.

No major changes were deemed necessary for the screens or the lenses, the screens and their lenses remain close to the user's eyes but beyond the spacing no steps were taken to ensure security at this stage. Through testing we found our platform, the Zero W, was not powerful enough to run additional sensors.

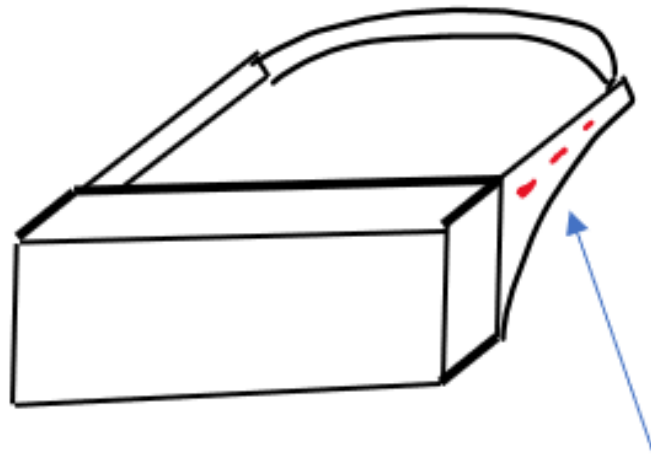
As mentioned before the Raspberry Pi Zero W is not a powerful platform. Currently the Pi struggles with running the screens and the testing software. For demonstrating basic feasibility this is fine, for larger scale production or more advanced development a more powerful Pi will be necessary. Using a Pi 2 or later generations has been suggested for

future platforms. The only difference between all platforms is the casing needs to be designed for different spaced mounting points and small changes need to be made in the boot process for the screens.

Secure data transfer is a major part of this project but is not something we currently have implemented. Our way of taking care of data security was originally going to be through the use of a VPN. We were going to use the Cisco Anywhere Connect VPN to connect to the ISU servers. Due to a connection issue inherent to the VPN, the VPN service can not be used on any Pi platform, unless a different VPN is used.

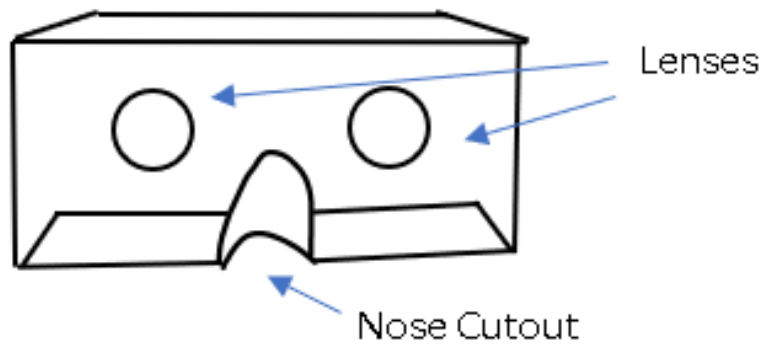
Perhaps one of the larger changes from the first semester design is the complete removal of any charging or battery capability. It was decided that there were other challenges in the design that required more focus than any ability to move around.

Front View



Buttons/controls

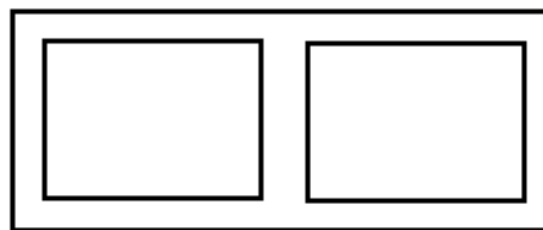
Back View



Lenses

Nose Cutout

Behind Lenses



Displays

Figure 3.3.1: Sketch of headset design

3.4 Technology Considerations

Most of the existing wearable technology out there today consists of VR and AR Headsets. A lot of the VR headsets on the market currently are highly focused on the gaming industry. Software could be designed to proctor a test using a VR headset. Though these devices are slowly reducing in price, most are still hundreds to thousands of dollars. As these devices are still expensive for the user and additional cost would be incurred designing testing software it has been deemed existing products are infeasible to design around.

Currently AR headsets are still under development, and few choices are out on the market. The ones that are on the market are thousands of dollars. In a similar manner, there are very few head mounted displays or other wearable displays currently available on the market.

There are existing remote testing solutions out there. The most popular type is a lockdown browser. Using a lockdown browser allows for students to use their existing computer to take the test, but work arounds are out there that allow students to still gain an unfair advantage, and to share the information from the test with others. A test taker could also use another external device to record the screen and capture test data. Such a gap in exam integrity is the reason why this project was put forward.

3.5 First Semester Design Analysis

The minimum viable product includes the following features: controls, display, data security, data transfer, power, and headset housing. The controls will be a scroll wheel and a few buttons. Data security will be accomplished using encryption libraries and tamper evident techniques for the hardware. Data transfer will be via wifi (or cellular for a future iteration). The small battery will ensure enough power to successfully transfer test data to the server in case of the headset not being plugged in and draining its power during an exam. The headset housing will hold all of the hardware and power equipment and will be built to be durable and comfortable.

After our initial analysis, we removed sensors as a component to ensure that a finished product can be produced in the allotted time. Although the minimum viable product will not be able to monitor the environment, future iterations may include sensors such as cameras, lidar, sonar, infrared, or microphones. Sensors and cameras would be used to ensure that the user was not cheating during the test. Sensor data would either be saved for processing by humans or AI could be implemented to tag specific moments of expected cheating. This AI is currently being developed in industry by certain proctoring companies, but is beyond the scope and expertise of our team and project. Future iterations may also include the following additional features: drawing tablet, built in calculator, eye tracking, hand tracking, and a larger battery. The drawing tablet will allow the user to write down work while still being within the head mounted display's environment, which minimizes the user's ability to cheat. The built in calculator will allow the user to do math calculations, which will increase the market for the product. Eye tracking and

hand tracking would enable the user to interact with the test in a different way than the headset buttons and scroll wheel. The larger battery would allow the user to take a test without being connected to power the entire time.

3.5.1 Second Semester Updates to Design Analysis

We have met four out of the 6 original MVP requirements, through further analysis of our problem that 6 initial requirements drop to 5. This is because through communication with our client batteries or charging solutions were not needed in the minimal viable product.

3.6 Development Process

We intend to follow the scrum methodology as it's what we're all familiar with and is the cycle we've already fallen into with product design. We have a meeting every Monday recapping what we've done in the past week and reviewing class assignments due in the coming week as well as questions for our industry contact meeting to address on Tuesday. Tuesday we ask those questions and work to further decide what needs to be done in the coming week and divide that up. This cycle seems to work well for the team and in these unpredictable times it is beneficial to have structure you can rely on.

3.7 Design Plan

Our product first and foremost has to get the test from the server, convey it to the user, and take input. That is demonstrated here by the software diagram in figure 3.7.1, which shows data first moving from the server to the headset wifi module, interfacing with Tao, running through a Tao reader, receiving input from the user and then going backward through the process to submit the test taker's input. The main restrictions on our design are comfortability and security. All the components are kept at the front of the head to keep everything securely contained and simplify the strap on the users head. Keeping everything sealed allows for the possibility of tamper detection in future designs, and by strictly limiting the input and output, we minimize the possibility of secure data getting into the wrong hands or inaccurate/malicious data coming in.

Justification for our first semester design plan is included below

Use Cases:

- Students in college or high school taking multiple choice or standardized exams
- Professionals taking certification exams

Justification for our use cases:

- Because students are a large portion of people taking exams, we are placing a priority on producing an affordable product.
- As certification exams and college entrance exams involve test data holding high monetary value, our product needs to be very secure when transferring data between the external server and the headset. We do not want users to be able to tamper with the

hardware and intercept any of the test data. As these types of exams are also relatively high stakes, we need to ensure that the users are not cheating. Future iterations of the headset will be able to detect when the headset is removed and at that point will pause the exam and not allow previous questions to be answered. Future iterations may also provide sensors or cameras that can support proctoring or monitoring of the exam environment.

Requirements:

- We need a way to select an answer as well as progress to the next question, so our design plan involves buttons on the headset that will allow user interaction. We are only supporting multiple choice questions for the minimum viable product, so buttons and a scroll wheel will be sufficient for user interaction.
- We need a way to communicate the exam to the headset, so we will be using an on chip wifi module built into the Raspberry Pi. This will allow the headset to connect to the internet and receive the test.
- We need to ensure that the headset remains charged throughout the duration of the exam, so the headset will be plugged in at all times for this first iteration. Future iterations may support a power supply that enables disconnecting the headset for a period of time.
- Because we need to ensure secure content transfer, we will be using encryption techniques.
- The headset needs to be cheat-resistant and cheat-evident, so we will implement a technique in the hardware that makes it obvious that the headset was tampered with. This may be a seal or disposable fuse.
- The headset complies with Question & Test Interoperability (QTI) Standards. We will be implementing the headset using Tao Software, which already complies with QTI Standards.

Modules, Dependencies, and Constraints:

The modules involved in the software architecture include a program to communicate with the external server, software similar to Tao as the backend to handle the data that is input, software to display the exam on the two screens, and software to allow the buttons and scroll wheel to interact with the display and Tao module.

Each module is dependent on the other adjacent modules. The external server has to communicate with our first layer of software. The speed of this communication is constrained by a user's wifi connection. To limit the negative effects on the user's experience while taking the exam, this first layer of communication from the server will happen before a user takes the exam. The test data will be downloaded from the external server onto the headset for later use to take the exam. The software that receives the test data then communicates with the Tao module, which stores the data. The Tao module is constrained by the on chip memory, as this will determine how large or a test the headset can store. That data then is used in a specific

layout that is displayed to the user on the two screens. The display clarity is limited by the pixels of the screens used. The user can interact with the display by using the inputs, which can select an answer that can be saved back in the Tao module.

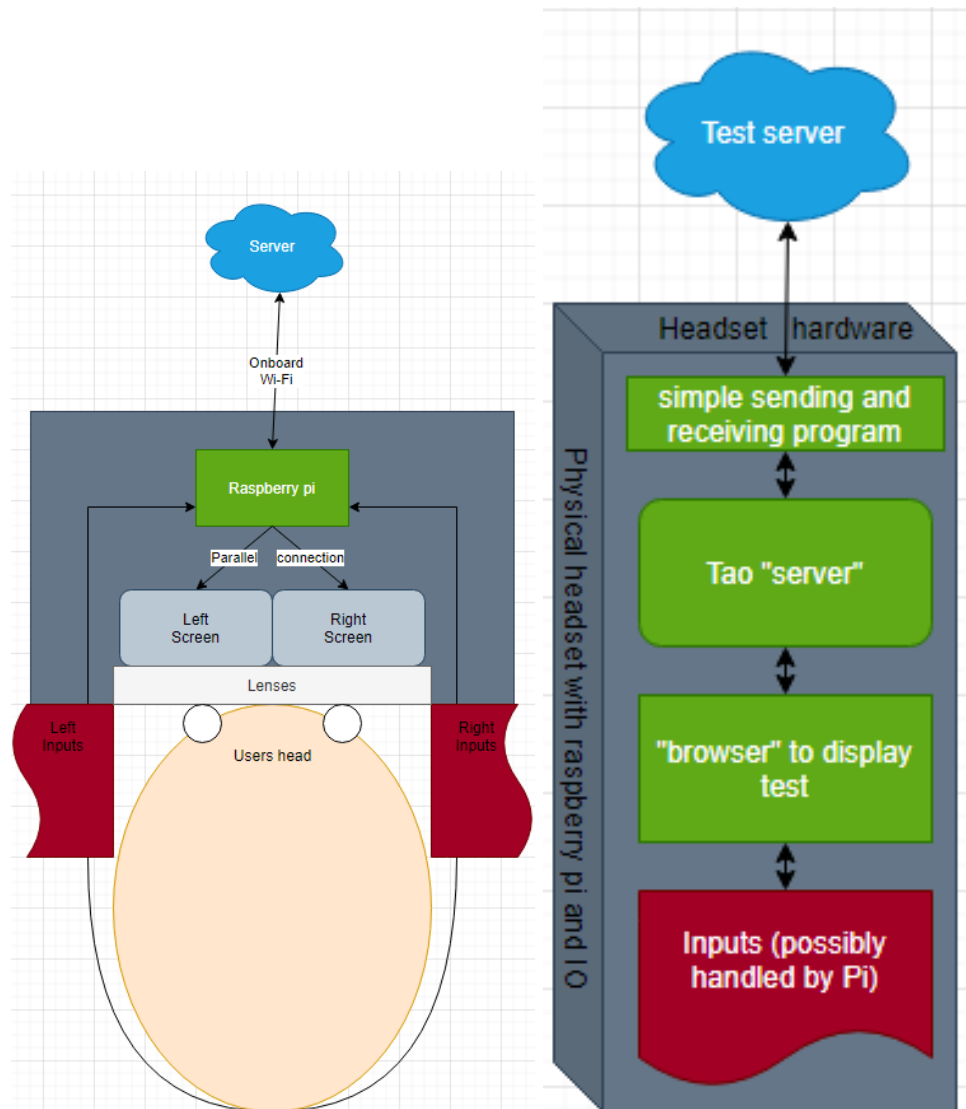


Figure 3.7.1: Hardware and software diagrams

3.7.1 Second Semester Updates to the Design Plan

There were a few changes to our design plan, all minor changes. The first change was in the input requirement, input will not be through buttons on the side of the headset. It was decided a hand held device would be more comfortable and the difference in security is negligible.

4 Testing

4.1 Unit Testing

4.1.1 Unit Testing Plans

Our unit testing plans involved hardware and software. The main components are included below.

Hardware Unit Testing:

- Computing system: The main chip that runs the program can output consistent signals that are expected test signals.
- Input functionality: Checks that inputs can send a signal
- Screen: Checks that screen receives power and displays correct image
- Other Components (Unknown at this time): Checks that all receive consistent power and send consistent signals.
- Wifi/Cellular: Component receives power and can send a signal.

Software Unit Testing:

- Data Transfer: Ensure that data is successfully transferred from host to device without data loss.
- Data Security: Ensure successful encryption of data that's transmitted from host to client devices.
- User Interface: Ensure basic functionality of user navigation in interface.
- User Input: Ensure that signals received from the hardware end are displayed correctly.
- Data Display: Ensure that data is successfully delivered to the interface and is displayed to the user.
- Wifi/Cellular: Ensure that the device is able to wirelessly connect, send data, and receive data.

4.1.2 Unit Testing Execution

Our implemented tests are included below.

Raspberry Pi Software

- Cornerstone Strategies logo appears upon boot in place of the default Raspberry Pi boot screen (SUCCESS)
- Raspberry Pi opens Chromium in kiosk mode once the boot sequence is complete (SUCCESS)
- Raspberry Pi opens the network name input screen upon boot if no network connection is detected (SUCCESS)
- Raspberry Pi redirects to the tiled navigation screen upon boot if an active network connection is detected (SUCCESS)
- After completion of network credentials forms, wpa_supplicant.conf file is updated with the entered credentials (SUCCESS)
- When wpa_supplicant.conf is updated, the network password in the new entry is WPA encrypted (SUCCESS)
- Upon update of the wpa_supplicant.conf file, the Raspberry Pi is automatically rebooted (SUCCESS)
- Upon reboot of the Raspberry Pi after entering valid network credentials, a functional network connection is acquired (SUCCESS)

4.2 Interface Testing

Computing system to screen testing: Screen can display an image loaded from the computing system and can switch to any required image as required.

Input to computing system testing: The inputs send a consistent signal to the computing system and the computing system can reliably interpret the inputs into usable data.

Image adjustment dial to screen testing: The dial can move the focal point for the user closer or farther from the screen with stability and can show an accurate image when movement is complete.

Computing system to Wifi/Cellular module testing: The computing system can send the data to the wifi/cellular module, and the wifi/cellular module can send error messages to the computing system.

4.2.1 Interface Testing Plan

This section includes a review of our testing plan.

- Computing system to screen testing: Screen can display an image loaded from the computing system and can switch to any required image as required.
- Input to computing system testing: The inputs send a consistent signal to the computing system and the computing system can reliably interpret the inputs into usable data.
- Image adjustment dial to screen testing: The dial can move the focal point for the user closer or farther from the screen with stability and can show an accurate image when movement is complete.
- Computing system to Wifi/Cellular module testing: The computing system can send the data to the wifi/cellular module, and the wifi/cellular module can send error messages to the computing system.

4.2.2 Interface Testing Execution

Our interface testing consisted of raspberry pi to screen, controls to raspberry pi, and raspberry pi to remote server.

Raspberry Pi to Screens

- The desktop display of the Raspberry Pi is duplicated and displayed identically on each screen (SUCCESS)

Controls to Raspberry Pi

- Handheld controller can perform all actions of a typical computer mouse (SUCCESS)

Raspberry Pi to Remote Server

- Raspberry Pi can connect to various internet URLs when connected to and within range of a WiFi connection (SUCCESS)
- Raspberry Pi can log into our Tao testing server when a user enters in valid credentials (SUCCESS)

4.3 Acceptance Testing

4.3.1 Acceptance Testing Plan

Our acceptance testing plan outlines end-to-end testing and client involvement:

- End-to-end testing
 - The requirements of the prototype will be tested rigorously from a use case perspective
 - All necessary use cases must be verified
 - Example use cases:
 - User can open an exam using the headset
 - User can submit answers
 - User can navigate between questions
 - User can submit an exam to be graded
- Client involvement in acceptance testing
 - The client will have authority on what use cases are necessary.
 - If there are fail points, the client will determine if they invalidate the acceptability of the prototype, and which fail points are most important to fix.

4.3.2 Acceptance Testing Execution

Acceptance testing execution includes the following components:

- User can connect to a nearby WiFi network by entering the appropriate credentials (SUCCESS)
- User can access various URLs from the navigation screen (SUCCESS)
- User can log into Tao by entering the appropriate credentials (SUCCESS)
- User can open and take a test from the headset after logging into Tao (FAILURE)
 - Test could not be loaded from the headset due to slow performance
 - Memory usage exceeded 100%, so swap space was being used, which caused excessive slowdown

4.4 Results and Possible Improvements

Our results and improvements based on executed tests are included below.

Technical Failure

- A Raspberry Pi 3B+ would have more onboard main memory and should enable the user to take a full test using the headset. We were not able to test this hypothesis because we did not have enough time.

Comfort Improvements

- The user interface is not suitable for comfortable use
 - The color scheme of the network credentials pages is harsh and uncomfortable to look at
 - The graphical keyboard on the network pages is too small for comfortable use

5 Implementation

Our prototype implementation uses a Raspberry Pi 0W microcomputer, along with two 240x320 IPS LCDs with a ST7789VW driver chip. First we installed Raspbian on an SD card and inserted it into the Raspberry Pi 0W. For development purposes, we connected the Raspberry to an external monitor to continue with initial setup and development. In order to achieve screen functionality, we followed the instructions mentioned in the *Setup Pi* appendix section, and performed the proper cmake command to allow for the screens to operate with our Pi. To ensure that users only had access to desired content, the Pi was set up in kiosk mode to boot into a browser on startup, where a basic network configuration HTML file would be served to them if they were connecting to a Wifi network for the first time. To allow for a user to connect to a Wifi for the first time, we modified the permissions of the `/etc/wpa_supplicant/wpa_supplicant.conf` file to allow for writing. Writing to this file would occur when a user enters the proper network information, at which point the Pi would configure the network information, and reboot into the kiosk mode browser on startup. To ensure security of new network information being delivered, we added password encryption into the configuration process. Once a user has network configuration complete, they see a page with a list of links that would redirect them to the corresponding site. Currently, the prototype has links to Google, example.com, an online multiple choice test, and a Tao testing server that we set up with the Iowa State VM. Going forward, these links would only allow the user to access sites where they would be able to find their testing content. Once a user navigates to the Tao Testing server, they can enter the proper credentials and take an example multiple choice exam that showcases the prototype's functionality. Our implementation uses a handheld mouse to allow for the user to interact with the display content .

6 Closing Material

6.1 Conclusion

In conclusion, our initial goal with this project was to develop a wearable device that would allow users to take a test on, help provide a more secure testing experience, and serve as a platform in the future for further sensors to fully allow the security of an exam. Our team believes that we have accomplished a prototype of this product demonstrating that for a low cost a device can be constructed and used.

Through this year, we ended up spending \$191.05 for the development of our prototype. This includes spare parts and tools, so taking out those extra expenses, we believe the product could be produced for around \$116.86. This does not include potential further saving from bulk pricing on parts, and the use of a better production method for the casing. Current pricing on the casing was based on the material used to 3D print it. Further details on expenses can be found in Appendix 6.

Through testing, we've determined that this prototype doesn't offer an experience good enough for a test taker. Due to the hardware used, a raspberry pi zero, the experience is slow and would hinder test takers with a time limit on their exam. In addition, in its current design, the device would only allow exam questions that can be answered with the use of a mouse based control, such as multiple choice questions. The team believes that with more capable hardware, this experience could be improved to the point where this could be a viable option to deploy tests on.

6.2 References

- “Average Peak & Median Internet Speeds USA”, fastmetrics.com. [Online]. Available: <https://www.fastmetrics.com/internet-connection-speed-map-usa.php/>. [Accessed: 12-Oct-2020].
- B. Ackroyd, “A Guide to Mobile Download Speeds: 2G, 3G, 4G and 5G. What Do They Actually Mean?,” *TigerMobiles.com*, 23-Jan-2019. [Online]. Available: <https://www.tigermobiles.com/faq/mobile-download-speed-guide/>. [Accessed: 27-Oct-2020].
- “Do you even need to wear glasses with VR?: Find out if you should be wearing your glasses with your VR headset.” VR Bound. [Online]. Available: <https://www.vrbound.com/guides/which-virtual-reality-headsets-can-i-wear-with-glasses>. [Accessed 15-Oct-2020].
- GitHub. 2019. *Release TAO 3.3.0-RC02 · Oat-Sa/Package-Tao*. [online] Available at: <<https://github.com/oat-sa/package-tao/releases/tag/v3.3-rc02>> [Accessed 15-Nov-2020].
- Jay. “How Lenses for Virtual Reality Headsets Work.” VR Lens Lab. [Online]. Available: <https://vr-lens-lab.com/lenses-for-virtual-reality-headsets/>. [Accessed 15-Oct-2020].
- “State of Mobile Internet Connectivity 2018.”, gsma.com. [Online]. Available: www.gsma.com/mobilefordevelopment/wp-content/uploads/2018/09/State-of-Mobile-Internet-Connectivity-2018.pdf. [Accessed 12-Oct-2020].
- T. Cooper, “Report: US States With the Worst and Best Internet Coverage 2018,” *Broadband Now*, 09-Mar-2020. [Online]. Available: <https://broadbandnow.com/report/us-states-internet-coverage-speed-2018/>. [Accessed: 27-Oct-2020].

6.3 Appendices

Appendix 1: Operational Manual

1. Power on headset using the power button attached to the chord (Fig. 6.3.1).



Figure 6.3.1: Off/on button for headset

2. Enter network information (Fig. 6.3.2 and Fig. 6.3.3).

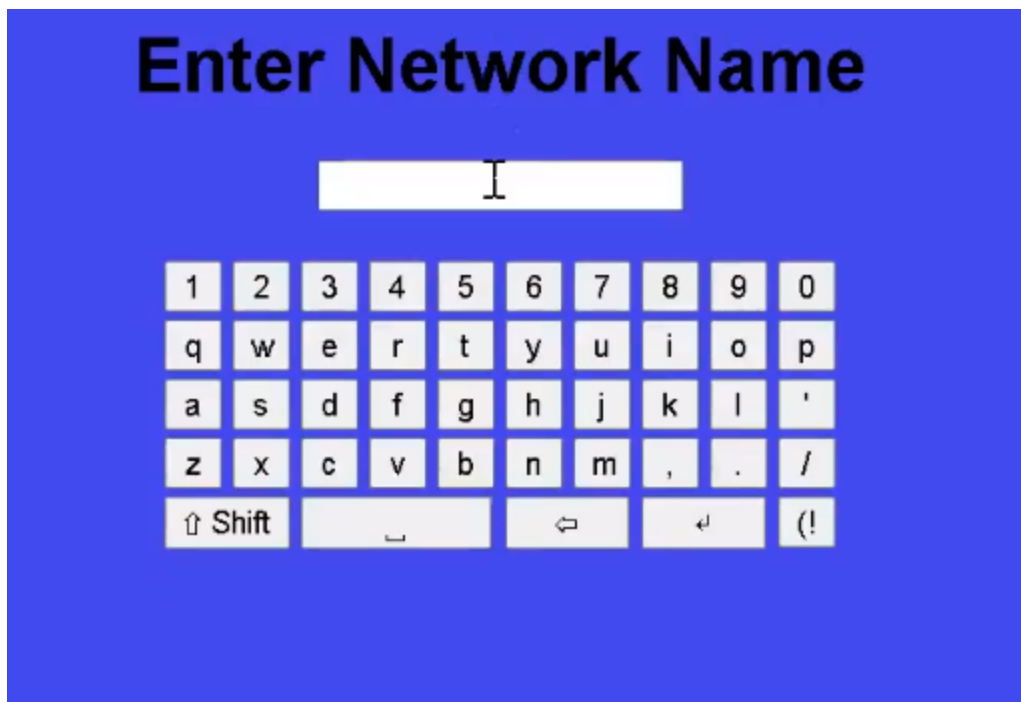


Figure 6.3.2: UI for entering network name

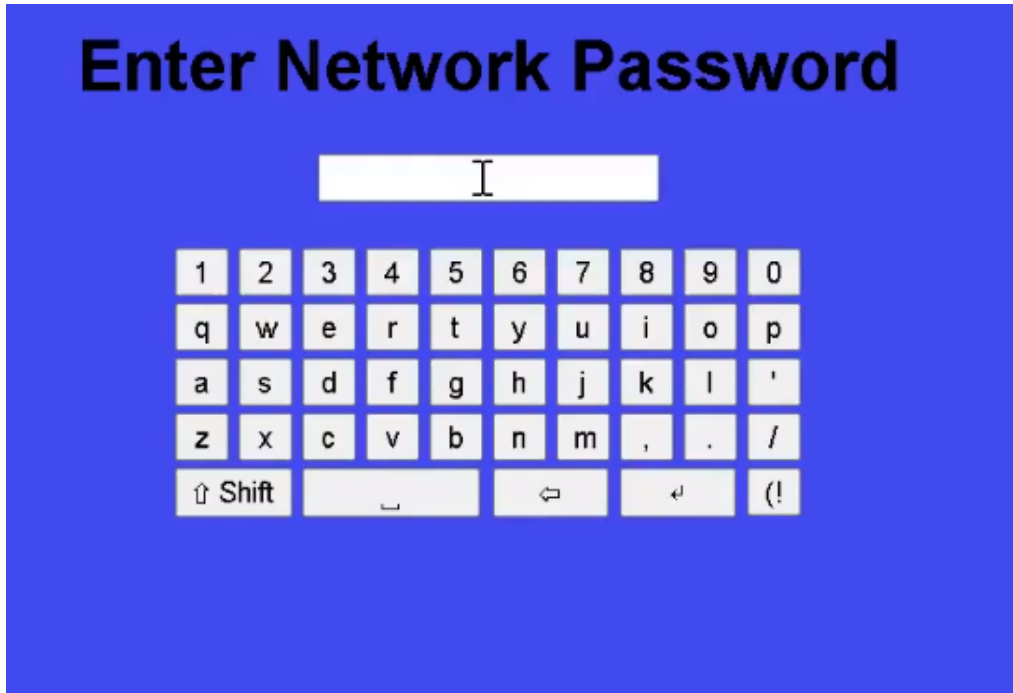


Figure 6.3.3: UI for entering network password

3. Navigate to the test driver of choice (Fig. 6.3.4).

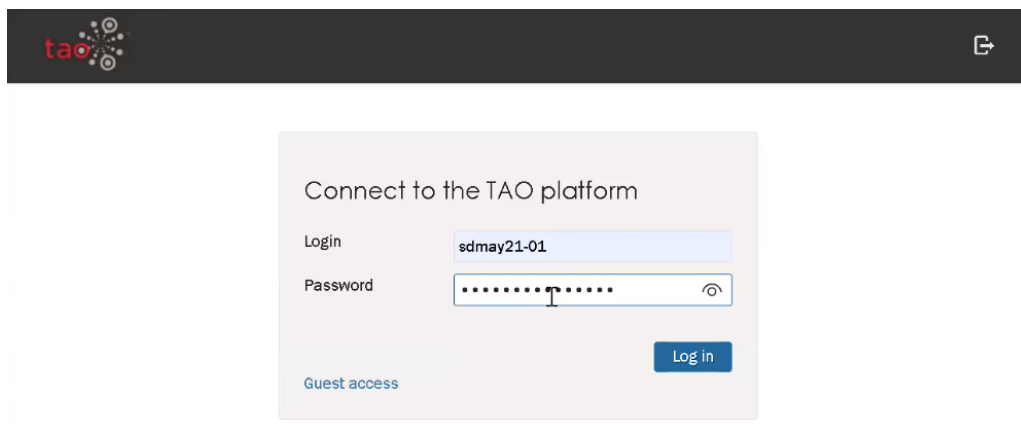


Figure 6.3.4: Tao platform login page

4. Complete exam.
5. Power off and on the headset to take another exam. Use the same power button used in step 1. Network information will be remembered after the first time, so the user will be taken directly to the test driver landing page.

Appendix 2: Related Product - Google Cardboard

Google Cardboard brings immersive experiences to everyone in a simple and affordable way. Whether you fold your own or buy a Works with Google Cardboard certified viewer, you're just one step away from experiencing virtual reality on your smartphone.

“Get Cardboard,” Google. [Online]. Available: <https://arvr.google.com/cardboard/get-cardboard/>. [Accessed: 16-Nov-2020].

Appendix 3: Related Product - Project North Star

At Leap Motion, we envision a future where the physical and virtual worlds blend together into a single magical experience. At the heart of this experience is hand tracking, which unlocks interactions uniquely suited to virtual and augmented reality. To explore the boundaries of interactive design in AR, we created Project North Star, which drove us to push beyond the limitations of existing systems.

“Project North Star is Now Open Source,” Leap Motion Blog, 06-Jun-2018. [Online]. Available: <https://blog.leapmotion.com/north-star-open-source/>. [Accessed: 16-Nov-2020].

Appendix 4: Design Decisions

Security:

- Hardware security: Ensure no physical tampering with device inputs
 - disable hdmi in config file (/boot/config.txt)
 - Snap off unused gpio pins
 - Hot glue SD card into the raspberry pi
 - Make case difficult to open, tamper evident
 - Use bluetooth input device to ensure data is encrypted and is more limited in range (cannot be intercepted beyond about 10ft instead of 50ft with the existing RF communication technique in the prototype)
- Software security:
 - Do not allow user to switch tabs or go to unwarranted sites
 - Chromium kiosk mode disables url bar
 - Do not allow VMs
 - Chromium kiosk mode disables access to the terminal
 - Ensure user cannot ssh
 - Disable ssh in config file
 - Ensure the correct user is taking an exam
 - (To be determined for future iteration - potential option is to use facial recognition or image of user that proctor can confirm)
 - Ensure user is taking an exam on the headset

- public/private keys, mac address, ip, token, (to be implemented in future iteration)
- Headset Casing:
 - Headset should weigh under 400 g
 - To comply with user studies on headset comfort
 - Strap around the head and over the top
 - Provides additional support, user studies have found this to be the most comfortable type of straps
 - Headset front casing split into two parts
 - Allows for easier assembly of hardware parts into the headset casing
 - Headset casing front bottom portion has to be large enough to allow for raspberry pi and GPIO header board along with connector cables
- Hardware:
 - With the possibility of GPIO input being needed, having the GPIO also do output was deemed better than HDMI output. This is because serial peripheral interface (SPI) would have to be enabled for HDMI output, and it did not make sense to have additional ports open.
 - In-Plane Switching (IPS) displays
 - No obvious difference between IPS and thin-film-transistor (TFT) displays so pricing was more of a factor than performance in this decision
 - No header mountings for screen
 - Discarded header mountings as two screens were needed, headers prevented this.
 - Two screens (one per eye) instead of one larger screen
 - Screen splitters would complicate the design. Just having two screens and not using a splitter is a simpler and likely more robust system.
 - Allow user to turn on and off the headset
 - Power switch used
 - No software technique to power down the pi without leaving kiosk mode which would become a security concern
 - Raspberry Pi Zero W
 - Initial design phase placed comfort, and as such weight, as a priority. The Pi Zero is the smallest model.
 - Wifi functionality was also required so the Zero W variant specifically was chosen.
 - Pi Zero allows for development of background programming with easy transfer to different Pi platforms. Later updates to the design should probably use a more powerful Pi such as a 3 or 4.
 - Pi Zero has few access points. When coupled with a sealed casing and disable ports, this provides sufficient security for our purposes

Appendix 5: Setup Pi

- Insert SD card with Raspian installed on it
- Connect micro usb to usb cable wireless handheld controller
- Connect pi to screens (or for development purposes connect mini hdmi to hdmi cable to a monitor)
- Connect power
- Run `sudo apt update` and `sudo apt full-upgrade`
- Install NodeJS
- Install formidable: `npm install formidable`
- Install cmake
- Follow instructions on <https://github.com/juj/fbcp-ili9341/blob/master/README.md>
 - cmake line is: `-DST7789VW=ON -DGPIO_TFT_DATA_CONTROL=25 -DGPIO_TFT_RESET_PIN=27 -DGPIO_TFT_BACKLIGHT=18 -DSPI_BUS_CLOCK_DIVISOR=10 -DSINGLE_CORE_BOARD=ON -DARMV6Z=ON -DSTATISTICS=0 ..`
- Change permissions on `/etc/wpa_supplicant/wpa_supplicant.conf` to allow write
- Installed Screen recorder for demos: `sudo apt install simplescreenrecorder`
 - Application can be reached from Raspberry icon then 'Sound and Video'
- Setup reboot of pi when network info added
 - Network.js:
 - `const { exec } = require('child_process');`
 - `exec('sudo /sbin/shutdown -r now', function(msg) { console.log(msg); });`
- Run file on reboot
 - Create shell script with the following code: `'node /home/pi/server.js &'` and put that shell script at `/etc/rc.local`
 - Move application files to `home/pi`, `sudo` not needed when ran from `home/pi`
- Open browser on startup:
 - Add to `/etc/xdg/lxsession/LXDE-pi/autostart`: `/usr/bin/chromium-browser --noerrdialogs --disable-infobars --kiosk --disable-restore-session-state http://localhost:8080/netset1.html`
 - Alt + F4 key combination to exit full screen mode (for developer purposes only, product user will not have keyboard access)
- Remove boot up screen and raspberry pi screen:
 - `/boot/cmdline.txt`: change `'ttyt 1'` to `'ttyt 3'`, ensure `'quiet splash'` is a parameter in this file
 - Put `'disable_splash=1'` into `/boot/config.txt` for removing rainbow
 - Replace the file `/usr/share/plymouth/themes/pix/splash.png` with custom logo (renamed as `splash.png`) or just remove it for a blank screen
- Install vpn client:
 - steps followed:
 - Install the vpn application: **`apt-get install openconnect network-manager-openconnect-gnome`**
 - Run the vpn with username: **`openconnect -u netid -b vpn.iastate.edu`**

- This will prompt for the password for the given user id
 - It will ask for a group, list **Primary**
- This does not work for ISU because of university MFA. You will end up in an infinite loop of asking for credentials. Although it may work for other VPN networks.

How to Attach Alternate Controls:

Step 1: Open Casing and Remove Screws

Open the headset casing and remove the four screws that connect the two front pieces of the headset casing that houses the hardware.



Figure 6.3.5: Initial Headset

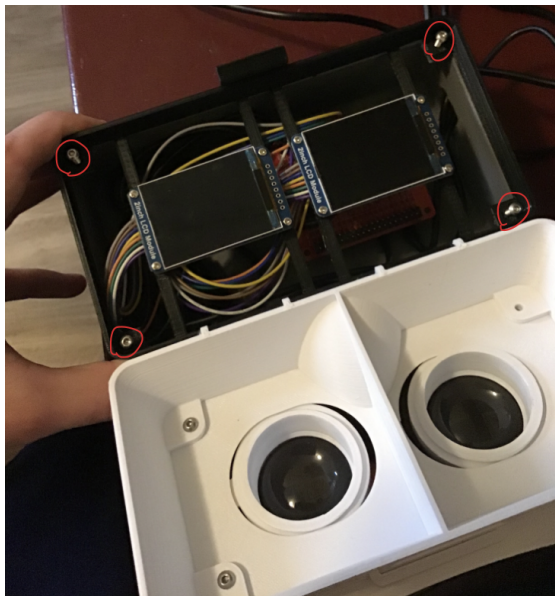


Figure 6.3.6: Open case with screws

Step 2: Rotate Top Portion of Front of Headset Casing

Rotate the top portion of the front of the casing clockwise, exposing the Pi and its GPIO header expansion board.

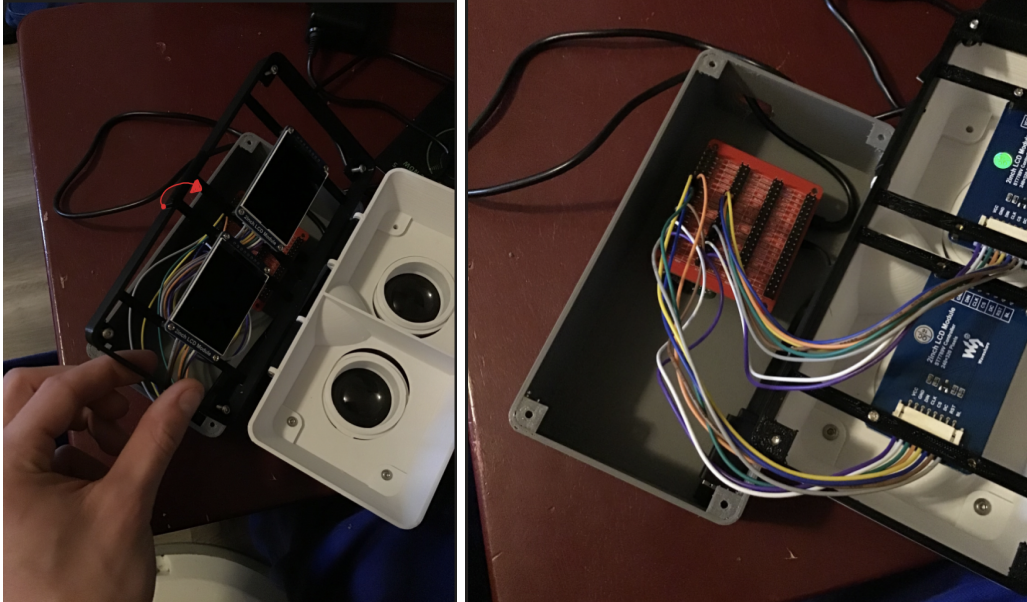


Figure 6.3.7: Open showing frame with screens **Figure 6.3.8:** Exposed GPIO Headers on Pi

Step 3: Remove Input Device Connection From USB Extender

Locate the USB Extender tucked along with the Raspberry Pi behind the screens. Disconnect the input device from the Pi by unplugging from the USB extender that's connected to the Raspberry Pi.



Figure 6.3.9: USB Extender with Dongle **Figure 6.3.610:** USB Extender with Trackpad

Step 5: Connect USB of New Input Device to USB Extender

Insert the USB of the desired input device into the USB extender that's connected to the Raspberry Pi.

Step 6 (If Necessary): Tuck Excess Cabling Into Empty Space of Front Headset Casing Portion



Figure 6.3.11: Example of Tucking Excess Cabling

Step 7: Rotate Top Portion of the Front of Headset Casing Counter-Clockwise and Align

Rotate the top portion of the front of the headset casing counter-clockwise so that it aligns with the bottom portion, as well as the four screw holes.

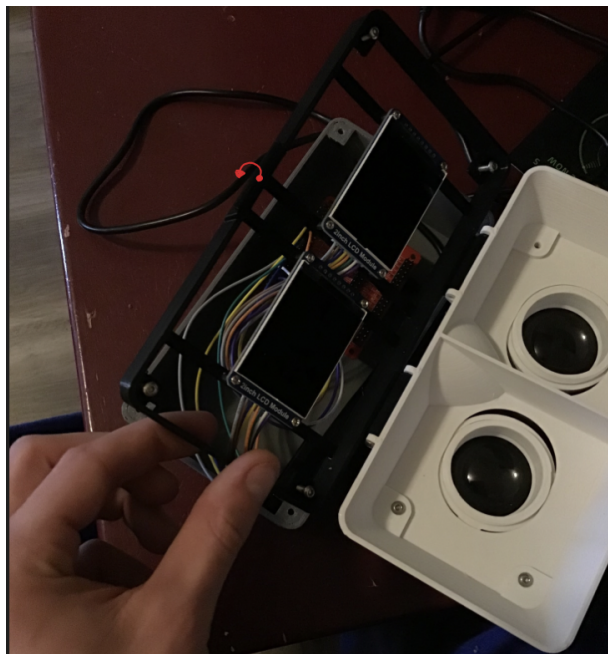


Figure 6.3.12: Returning Screen Frame

Step 8: Reinsert Screws and Close Casing

After aligning the two pieces of the front of the headset casing, reinsert the four screws and close the headset. Refer to figure 6.3.2 for in step 1 for screw locations. New input is ready to be used with the prototype.

Appendix 6: Future Iteration Features

- Input
 - Keyboard
 - Drawing tablet
 - Hand tracking
 - Tether handheld to headset to allow constant charging
- Sensors
 - Camera(s) looking externally
 - Camera watching eyes
 - Camera (with AI)
 - Infrared
 - Sonar/Lidar
 - Microphone
 - Gyroscope
- Software
 - Admin screens
 - Allow admin to add additional test drivers to landing page
 - Calculator app built-in
- Hardware/Headset Casing
 - Battery power
 - Potentially use a pi with more processing power or RAM if funds allow it
 - Cellular data transfer
 - Cut blue light to diminish eye strain
 - Further study on weight distribution and optimizing headset comfort
- Security
 - Encryption
 - Software to recognize if headset has additional devices added
 - Hand held:
 - randomize cursor location on boot up
 - screen wrapping (no borders can be recognized)
 - Use bluetooth handheld (more encryption is available for the data from handheld to device)
 - Headset:
 - how to recognize if extra devices connected
 - Tamper evident casing
 - Ensure that user is taking the test with the headset (as opposed to on a desktop computer) by recognizing something like its MAC address, ensure the person taking the test is who they say they are
- Other considerations
 - Accommodations for special needs (adjustable lenses for nearsighted/farsighted users)
 - Support
 - Send logs from headset

Appendix 7: Developmental Expenses

Bellow is a table showing our developmental expenses for this project:

Item	Quantity	Unit Price	Total Price	Estimated Bulk Price (seems to be about 17 % off when in bulk)
Raspberry Pi Zero W	1	\$37.30	\$37.3	\$26.11
Raspberry Pi Zero W	1	\$36.30	\$36.3	N/A* ^[1]
LCD Screen	2	\$10.95	\$21.9	\$15.33
Jumper Cables	1	\$1.95	\$1.95	\$1.37
Triple GPIO Expansion Board	1	\$7.95	\$7.95	\$5.57
Lenses	2	\$21.99	\$43.98	\$1.80* ^[2]
Foam for Casing	1	\$12.67	\$12.67	\$8.87
Headset Strap	1	\$8.00	\$8	\$5.60
Touchpad	1	\$21.00	\$21	\$14.70* ^[3]
Wireless Handheld	1	\$30.35	\$30.35	\$21.25* ^[3]
3d Print/Casing	1	\$46.20	\$46.2	\$21.00
Hardware & Tools	1	\$15.00	\$15	minimal* ^[4]
Power Switch	1	\$3.11	\$3.11	\$2.18
Shipping	1	\$37.89	\$37.89	
		Total Expenses:	\$191.05	\$101.16*^[5]
			Development	Product Estimate

Notes from table:

1. A second Raspberry Pi Zero W was purchased from a different supplier to assist in development. A second one is not needed for the final product.
2. Original lenses for development were sourced from amazon, but a bulk supplier was found through [aliexpress](#) with significantly reduced prices.
3. Only one control option is needed, though two were brought to test as potential options.
4. A kit was bought for development that had various screws, bolts, washers and tools included. Final product would only need a few screws for assembly.
5. Final Product Estimate was calculated only including the touchpad as the control option as that was the option that was preferred by the client.

Additional expense notes:

- Items unincluded: technical and customer support, packaging, additional tamper proof technology (physical/software security)
- Distributor option for bulk raspberry pi: Farnell and RS, 150 per box

- 3D print source option: <https://get.xometry.com/quote>
 - ABS/PLA material: avg 5cents/g