****

**Visual Voyagers**

Bringing Social Interaction to Virtual Reality

Through Visual Engagement

*Final Report*

Estella Chen

Citlali Haro Franco

Nick Nguyen

Annie Ren

Vivian Vo

**Department of Computer Science and Engineering**

**Texas A&M Engineering**

Fall 2023

**Table of Contents**

[**1 Executive Summary 2**](#_ssvan0xy02pg)

[**2 Introduction 4**](#_r9vv6rxpmf7)

[2.1 Problem Background 4](#_p1hgue85ue8p)

[2.2 User Stories 4](#_dq1ee4iu3qsz)

[2.3 Needs Statement 5](#_k41xoed3r2pw)

[2.4 Goals and Objectives 5](#_2c7to4kww23x)

[2.5 Design Constraints and Feasibility 6](#_jls2nrywy8ff)

[2.5.1 Hard Constraints 6](#_eq73hefk0688)

[2.5.2 Additional Features 7](#_quiy8k6bx1xm)

[**3 Project Design 8**](#_ymkllkoejq8z)

[3.1 Alternative Designs 8](#_jbduo2rjnhh)

[3.2 System Overview 10](#_c5i1bhsm84aq)

[3.2.1 Updated Wireframes 10](#_myjdl43y7ohj)

[3.2.2 Architectural Design/Backend Logic 12](#_v2qmwsck5t2d)

[3.2.2.1 Multiplayer Logic 12](#_xliy3gdhe99v)

[3.2.2.2 Google Photos API Logic 13](#_2c04m7e1txf0)

[3.2.2.3 Vivox Logic 14](#_hnibbq5pcb4e)

[3.3 Hardware 15](#_2xg5dn6g4iqv)

[3.3.1 HTC Vive 15](#_aslg88ugrvlz)

[3.3.2 Oculus Quest 2 15](#_je8a5aeunk8q)

[3.4 Limitations and Next Steps 16](#_7mgzfxc8lj47)

[3.4.1 Design Decisions, Changes, Omissions 16](#_8aa3pzvgjxpw)

[3.5 Measurements of Success 16](#_mtieyrehnk8h)

[3.5.1 Quantitative Measurements 16](#_xjw7mdvwrsvp)

[3.5.2 Qualitative Measurements 18](#_l0n23g8c793q)

[3.5.3 Validation Testing 19](#_dkizvazgwvue)

[3.5.3.1 Question Cases 19](#_8amcu1m2apjj)

[**4 Validation Testing Results & User Feedback 21**](#_h1c9xcn7h67c)

[4.1 Validation Test Results 21](#_m6gwphrb4o2i)

[4.2 Evaluation Process 21](#_1nth5fqbjqoh)

[4.3 User Feedback 21](#_4vq2qyo6pb6y)

[4.4 Alternative Product Solutions 23](#_hl8yp3ewe1sa)

[**5 User’s Manual 24**](#_r3thvqsu8zb2)

[5.1 VR User Manual 24](#_ws22hgberds3)

[5.2 Emulator User Manual 25](#_rkjmcfcvyjoc)

[**6 Engineering Standards 26**](#_ydgepkd659o2)

[6.1 Project Management 26](#_s5lvt26mylh2)

[6.1.1 Team Roles 26](#_lukzc6o42z9z)

[6.2 Updated Gantt Chart and Schedule of Tasks 28](#_2hayop5luvrb)

[6.3 Societal Impact 30](#_4mzdcgewpyl)

[6.4 Ethical Impact 31](#_51mvyso8wubt)

[6.5 Itemized Budget 32](#_3p8qsc3tk2w7)

[**6 References 33**](#_fotkbrb20dg)

[**7 Appendices 35**](#_l0inwx80op30)

[7.1 Resumes 35](#_c17us6kcc4zc)

## 1 Executive Summary

People lead increasingly busy lives in today’s fast-paced world. As people become engrossed in their schedules and routines, they drift apart and forget to keep in touch with their loved ones. Without thinking, people can quickly lose the connections they worked hard to build over time. Despite how interconnected our world is today with technology and digital photos to help remind us of what truly matters, people rarely take the time to look back through those photos.

While there are existing products that facilitate connection and allow people to look back on their memories, these alternatives are limited in their capabilities. Virtual reality (VR) allows for an immersive experience that can make users feel as though they’ve been transported into a different realm. Despite the constraints of physical boundaries, VR can create a feeling of closeness. VR can virtually place people next to their loved ones and encourage meaningful conversation.

This brings us to our solution, *Visual Voyagers*, which allows users to communicate with their loved ones in a local shared space. We opt to have users interact in small groups to encourage a stronger sense of connection. To further encourage this bonding, we emphasize the usage of “memories” or photos that will invoke positive past instances through various interaction methods. Users will be able to view and engage with their photos and memories and use that newly created experience to further strengthen their bond with their loved ones. Users can immerse themselves by viewing their photos in our virtual gallery, sharing this space with their loved ones through a multiplayer feature, and communicating with their loved ones through a voice chat feature.

Throughout and following the implementation of our project, design validation was conducted in order to meet our projected goals. Both quantitative and qualitative testing ensured that the application runs for at least 30 minutes without crashing or consuming too much battery all at once, reaches 90 frames per second (FPS) on a laptop and PC in order to have a satisfying user experience, and connects to the internet to enable our multiplayer feature. These technical aspects allow us to meet our user’s expectations.

Lastly, we have determined team roles and responsibilities, monitoring our progress with bi-weekly meetings and an Agile development methodology. Our project has many societal and ethical impacts such as enriching the user’s connections and relationships with their loved ones.

## 2 Introduction

### 2.1 Problem Background

When was the last time you had a conversation with someone you love? Have you been feeling disconnected from someone you were once close to? Has the distance between you and your loved one only felt like it was growing wider and wider? Well, you are in luck because we have just the thing for you!

We, Visual Voyagers, have tackled the issue of weakened connections by building a program that will bring a social space to the VR world. In this space, people can interact with their photos, or “memories”, in order to strengthen an existing connection with a loved one.

### 2.2 User Stories

Visual Voyager’s goal is to connect individuals with their loved ones through interactive activities that invoke memories via photos and shared experiences. The goal in itself can be hard to visualize or comprehend since we have to think of how we will be connecting users in addition to how to maintain the theme of engagement with visual elements like photos and images. Our core features will provide essential services to meet our users’ wants and needs by fulfilling specific requirements to enhance user satisfaction. To illustrate this more properly, we will convey the common needs through two separate user stories that detail different aspects we will target.

**Jason**

Our first user story follows Jason, a biology major attending college out of state. Being a biology major and third year in college, he is quite busy and finds it hard to find the time to socialize or relax. This ranges from meeting up with his friends from university to chatting with his old friends from his home state. Whenever Jason does find the time for leisure or there is a holiday break he prefers to go to events or trips with his friends, where he tries to take several pictures or videos to record their memories together. These recorded moments are all placed in a shared space where Jason and his friends can all view memories and photos of each other, walk through them together, and react to them. Jason finds this to be an engaging, fun, and meaningful way to connect with his friends and remain in touch with his older ones.

**Chris**

Our second user story follows Chris, an old, widowed grandfather. He lives by himself in a separate state from his immediate family and grandchildren. Recently, he’s been feeling rather lonely so he wants to try and connect to his young grandson Timmy. He wants to be able to connect with Timmy in a more meaningful and interactive way than just a phone call.

We believed in having a varied set of user stories to be able to convey all the aspects in which our product will be able to help people. These user stories showcase people of differing age groups, life phases, and backgrounds. It also highlights how our product addresses a need beyond a single target group, providing value to a range of people. The user stories also express how people want to connect and form stronger relations with their existing loved ones regardless of the distance between them. It also highlights the need to be able to interact with their loved ones via an engaging method that will be able to induce a more meaningful form of bonding.

In summary:

* People want to be able to **connect** across long **distances** with their loved ones**.**
* People want to build **stronger relations** with their loved ones
* People want to interact with **engaging**/novel **activities** while bonding.

### 2.3 Needs Statement

From the user stories we have determined the needs statements to satisfy different aspects of our users’ needs. We have observed that people desire connection. In order for people to connect with loved ones they would need a shared space to meet and interact in. We have also observed that people seek engagement thus we want people to be able to interact with their “memories” or photos in some meaningful or impactful way. Reminiscing on these memories, while it can be a lone task, is much more impactful when it is done between two or more people; therefore we want that added layer of interaction not just with “memories” but with people and their loved ones.

In summary:

* Users want a shared space to meet up and connect with their loved ones.
* In this shared space, users want to be able to reminisce on memories together in a meaningful way.

### 2.4 Goals and Objectives

Our main overarching goal is to create an immersive social space app in VR that focuses on connecting with others who are already known to the user and facilitates reminiscing on past images and memories. Additionally, there will be ways for users to interact with their images to increase engagement and foster meaningful relationships.

We have established a set of clear objectives:

* Our first objective is to create an immersive experience by pulling photos from existing albums and displaying them in VR.
* Our second objective is to facilitate a feeling of occupying a shared space in VR so users can join with their loved ones, see them in a first-person perspective, and be able to view each other’s movements.
* Our last objective is to incorporate a form of communication to enable users to share meaningful conversations and messages with each other within our app thus eliminating the need for a third-party app.

### 2.5 Design Constraints and Feasibility

For the constraints we have set some for ourselves in terms of design such as the performance of our program. We are dealing with an older model of the HTC Vive VR headset and in addition to that the program we are running is more resource intensive. This brings us to set a constraint for ourselves to have our program be able to achieve at least 90 FPS in order to have a satisfying user experience. Aside from the fact that the HTC Vive headset is not compatible with Macbooks, we have chosen to restrict the type of devices or hardware our program will run on to only laptops/PCs and to exclude less powerful devices such as mobile phones. This is to ensure that the devices have enough resources to run our program and achieve that 90 FPS.

Additionally, because our headsets are of an older model and we have developed a program that could be quite resource-intensive due to the graphics, some other concerns could involve the duration of our program and how long it can run without crashing or consuming too much battery all at once.

A constraint that is potentially out of our control is the connection to the internet which can prevent us from even using our multiplayer features, rendering the goal behind our project moot.

Some other details to take into consideration that are not exact constraints would be that our program may not be compatible with other headsets. This can limit the usage only to the HTC Vive line of headsets. Other minor factors could be user interface technicalities like the program not being intuitive to use.

#### 2.5.1 Hard Constraints

**Performance**

* Our app needs to achieve at least 90 frames per second (FPS) for an enjoyable experience when meeting the minimum hardware requirements.
* Our app needs to be able to run for at least 30 minutes without crashing or consuming too much battery all at once.

**Internet connection**

* Users will need an internet connection in order to use multiplayer.

#### 2.5.2 Additional Features

**Compatibility**

* We would like our app to be compatible with different types of headsets.

**User interface**

* An intuitive and user-friendly interface would improve the user experience.

## 3 Project Design

### 3.1 Alternative Designs

|  | **Google Photos** | **VR Photo Viewer** | **VRChat** | **Visual Voyagers** |
| --- | --- | --- | --- | --- |
| **Platform** | Mobile + Web | VR | VR + Desktop | VR |
| **Multiplayer or multiple users** | ✓ |  | ✓ | ✓ |
| **Immersive experience** |  | ✓ | ✓ | ✓ |
| **Photo uploading** | ✓ | ✓ |  | ✓ |
| **Photo viewing** | ✓ | ✓ |  | ✓ |
| **Social interactions** | Limited  (sharing) |  | ✓  (superficial) | ✓  (personal) |
| **Voice chat** |  |  | ✓ | ✓ |

**Figure 1.)** Table of various competitors and what features they

offer versus what Visual Voyagers plans to offer

**Google Photos**

The first alternative design we researched is Google Photos. This app allows multiple users with different devices to collaborate and upload images in a single shared album. However, when it comes to interactivity, Google Photos only allows users to like other images or leave comments on them. This limited interaction makes it difficult for users to strengthen bonds over long distances. Additionally, this app is viewed on a flat display, making it not very immersive (Google, 2023)[[11]](https://www.google.com/photos/about/).

**VR Photo Viewer**

Another alternative design we looked into is VR Photo Viewer. This app allows users to view photos stored in Google Photos or view locally stored photos in VR. Though it excels in this regard, it lacks in other areas that would promote connection such as multiplayer capabilities. Additionally, users cannot interact with the photos and the app is strictly for viewing only (SprocketVR, 2017)[[12]](https://store.steampowered.com/app/531980/VR_Photo_Viewer/).

**VRChat**

Lastly, we explored VRChat, which is a large VR social platform that facilitates global interactions between users. Within VRChat, users have access to a plethora of different activities, including concerts and mini-golf. Because there is a large online community, VRChat focuses on creating new connections and meeting other people. However, this does not align with the specific needs and expectations of the individuals featured in our user stories. Additionally, users are not filtered and are allowed voice and text chat, which would allow the use of inappropriate or abusive language. Users may also use a “modded client,” which would allow them to forcefully close other users’ games or ruin immersion in other ways (VRChat Legend Wiki, 2021)[[7]](https://vrchat-legends.fandom.com/wiki/Wiki_Policy_on_Hacking). Furthermore, VRChat lacks the ability to upload or view photos, which is a crucial feature sought after by our target users (VRChat Inc., 2017)[[13]](https://hello.vrchat.com/).

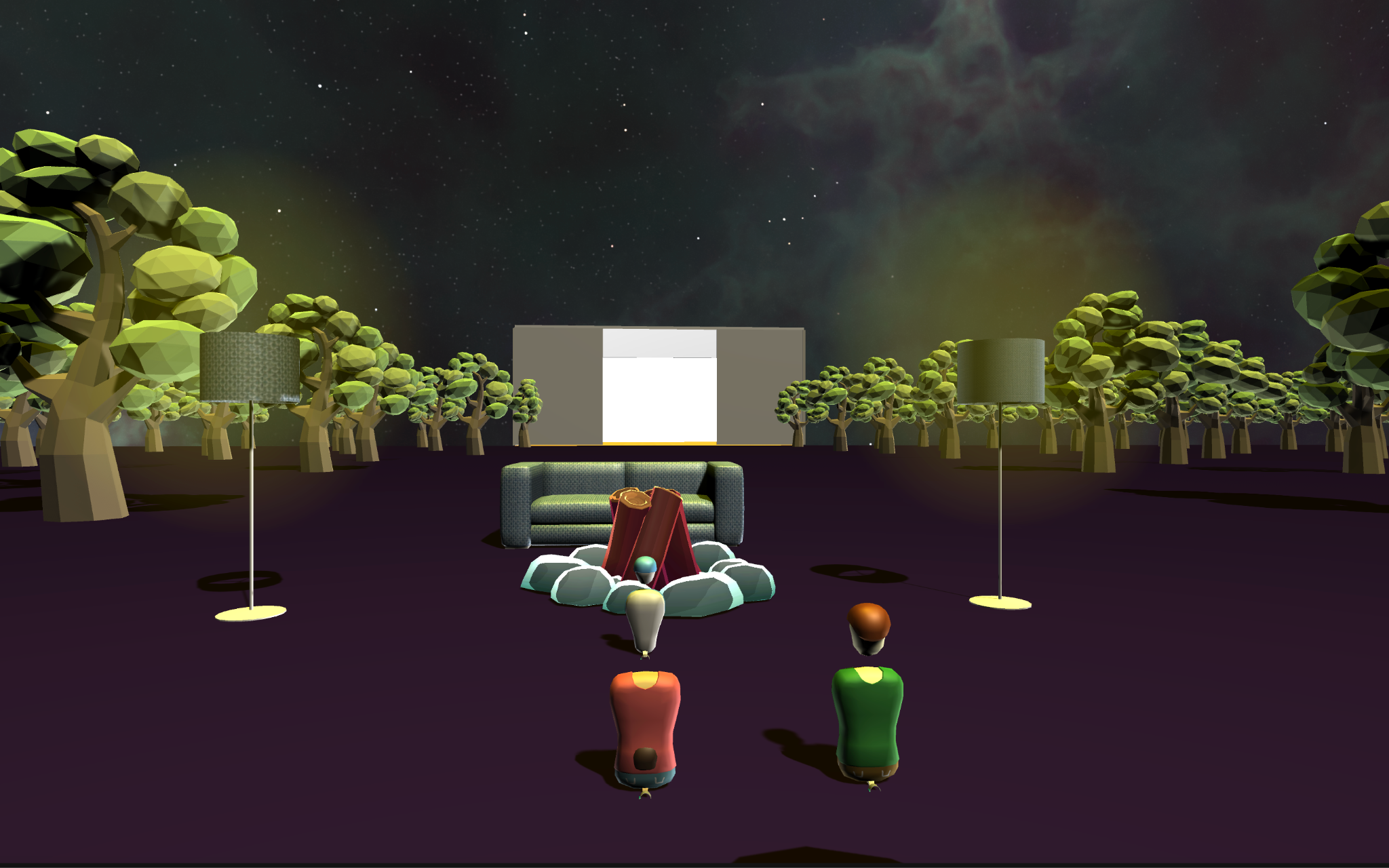
### 

### 3.2 System Overview

#### 3.2.1 Updated Wireframes



**Figure 2.)** Screenshot of outdoor scene



**Figure 3.)** Another perspective of the outdoor scene that shows the avatars and the gallery in the back.

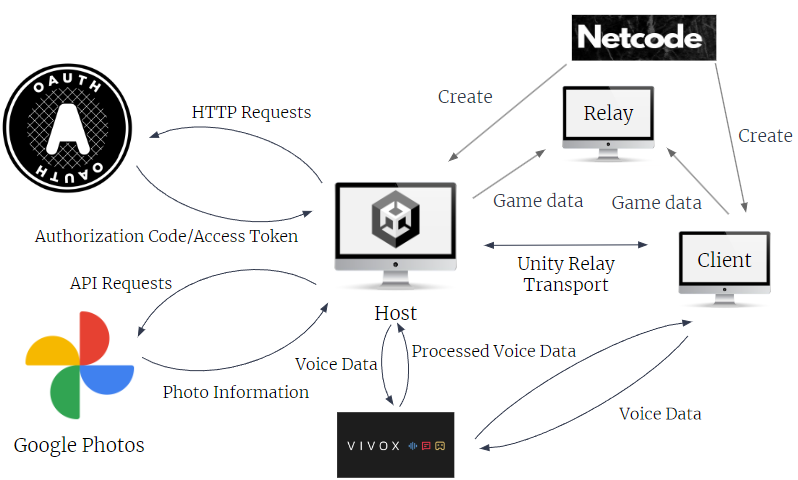


**Figure 4.)** Screenshot of inside the gallery space without photos



**Figure 5.)** Gallery space with textures applied on photos

#### 3.2.2 Architectural Design/Backend Logic



**Figure 6.)** Architectural Design

##### 3.2.2.1 Multiplayer Logic

To implement multiplayer functionality, Visual Voyagers uses Unity’s Netcode for GameObjects, which is a high-level networking library that allows you to send data to multiple players. We utilized Netcode for GameObjects by creating a Network Manager component that allows you to start a server, host, or client. The host acts as both a server and a client at the same time. To start a multiplayer session, the Network Manager uses an IP address and the port provided to a Unity Transport component to connect a client and a host (Unity)[[20]](https://docs-multiplayer.unity3d.com/netcode/current/components/networkmanager/). Game data, such as player actions or game state data, is then sent between the host and client through the UDP protocol (Unity)[[18]](https://docs.unity3d.com/Packages/com.unity.transport@2.0/manual/index.html). This allows Visual Voyagers to synchronize scenes between two players.

One limitation of our initial multiplayer implementation is that it only worked with devices on a local network. Since we have to connect to a host player’s IP address, if a client player is not on the same network, they could face problems, such as firewalls, and be unable to connect. To combat this problem, we implemented Unity Relay. Unity Relay allows players to connect to a Relay Server’s IP address and port. The players would then connect to the host player’s game session through a join code generated by the Relay API (Unity)[[19]](https://docs.unity.com/ugs/en-us/manual/relay/manual/relay-servers).

As we delved deeper into multiplayer integration, a significant challenge we faced was player synchronization. To address this, we transformed the local generic player into a network prefab, which allowed the Network Manager to handle the instantiation of the Generic Player prefab for network-connected players. We designed avatars with NetworkObject components for Unity Netcode management and created a PlayerNetwork script to handle instantiation for local, server, and remote players. This script ensured remote players would instantiate their own XR Origin Game Object locally, avoiding duplicate audio listeners in the session. In other words, the script removes the XR Origin Game Object in the game of remote players since it would be available locally for the remote player, allowing the client to have its own controllers locally. This way, we ensured clients had their own XR Origin input system setup.

Additionally, we adjusted the network connection script to accommodate Unity's concept of ownership. Players needed to control their own GenericPlayer, which meant assigning ownership and authority to the instantiated client and its controls. Proper assignment of ownership upon spawning was crucial. Therefore, we ensured that NetworkObject.SpawnWithOwnership() was called with the correct client ID on the server upon instantiation.

The integration process revealed that clients couldn't move their Generic Player without designated permissions. Unity defaults to server-only write permissions for network variables, which meant clients couldn't update their positions or initiate teleportation (i.e., clients were not able to write to their network variables of the client’s position, rotation, and teleportation). Therefore, the technical design of our project was adjusted to allow clients to write changes to their Network Game Object and inform the server, which then synchronized these updates network-wide. The Client Network Transform permitted the clients to process player input locally and replicate changes across the network via the server. The network instantiation of the Generic Player prefab by the Network Manager for the client's NetworkVariables was instrumental in synchronizing position, rotation, and teleportation of avatar components across players. Moreover, we employed state interpolation for smoother movement transitions to enhance synchronization and mitigate network latency.

##### 3.2.2.2 Google Photos API Logic

For the photo retrieval process, Visual Voyagers initiates user login through an in-environment call to action. When the user clicks the button, Unity sends an access request to Google's authentication server, OAuth. After users log in, they encounter a consent request. Upon consent, OAuth generates an authorization code with the requested permissions. This code is delivered to the client via a temporary locally hosted HTTP server on the user’s computer, and then it is automatically relayed back to Unity. The code will be used by Unity to initiate a server-to-server request to OAuth, which processes it and responds with an access token. This access token allows Unity to retrieve the user's data on their behalf, within the specified permissions (Google, 2023)[[15]](https://developers.google.com/identity/protocols/oauth2).

Using the access token, Unity submits API requests to Google Photos. If the access token is still valid, Google Photos will respond with the requested information, typically formatted as a JSON string. This JSON includes information such as the photo ID, creation date, author, and most importantly, the image URL. Unity creates an HTTP request using the URL to access the image. The image's texture is then downloaded into a buffer using Unity’s DownloadHandlerTexture. Finally, the texture is immediately applied to a scene component for real-time viewing within VR. (Google, 2023)[[16]](https://developers.google.com/photos/library/guides/access-media-items).

##### 3.2.2.3 Vivox Logic

In the communication process, Unity takes charge of capturing voice data, which is then sent to the online Vivox servers. These servers are responsible for routing, processing, and distributing the data to the intended clients. In our context, the other clients will be the users that currently have a connection with the host’s server. After the client applications successfully receive the data, users can engage in real-time audio conversations, listening and responding to one another (Unity, 2023)[[17]](https://docs.unity.com/ugs/en-us/manual/vivox-unity/manual/Unity/vivox-unity-first-steps).

### 3.3 Hardware

We used two VR headsets during development, the HTC Vive (2016 Model) and the Oculus Quest 2 (2020 Model).

#### 3.3.1 HTC Vive

The headset supports:

* 1080 × 1200 pixels per eye
* 90 Hz refresh rate
* 110-degree field of view
* adjustable lens distance

The headset comes with 2 controllers and 2 base stations that use Vive Base Station tracking and support multiple input buttons. The HTC Vive is not a standalone headset, and it requires a PC to run.

The minimum specifications of the PC required to run the HTC Vive are:

* Processor: Intel® Core™ i5-4590 or AMD FX 8350 equivalent or greater
* Memory: 4 GB RAM
* Graphics: NVIDIA® GeForce® GTX 970 or AMD Radeon R9 290 equivalent or greater
* OS: Windows 10

#### 3.3.2 Oculus Quest 2

The headset specifications are:

* Qualcomm® Snapdragon™ XR2 Processor
* 6 GB RAM
* 1832 × 1920 pixels per eye
* 90 Hz refresh rate
* 110-degree field of view
* adjustable lens distance with three settings: 58mm, 63mm, and 68mm (Moody, 2023)[[21]](https://studiox.lib.rochester.edu/oculus-quest-2-specifications/)

The headset comes with just 2 controllers. The Oculus Quest 2 is a standalone headset, so it does not require a PC to run, but can connect to one.

The specifications of the PC required to run the Oculus Quest 2 are:

* Processor: Intel® Core™ i5-4590 or AMD Ryzen 5 1500x equivalent or greater
* Memory: 8 GB RAM
* Graphics: NVIDIA® GeForce® GTX 970 or AMD Radeon 400 Series equivalent or greater
* OS: Windows 10 (Meta, 2023)[[22]](http://www.meta.com/help/quest/articles/headsets-and-accessories/oculus-link/meta-quest-link-compatibility/)

### 3.4 Limitations and Next Steps

#### 3.4.1 Design Decisions, Changes, Omissions

The main change we made since our Project Proposal was the decision to cut out games from our finished product. Through the course of making our application, we realized that there are a lot more moving parts to the multiplayer aspect of the application. We initially thought that implementing Netcode for GameObjects would be enough, but it only allows players on the same local network to connect to each other. We will be redirecting our efforts toward building on top of the multiplayer functionalities we have currently and implementing Unity Relay, which will allow for online multiplayer.

Following the implementation of Unity Relay, we were able to get player models to sync between the host and client, but we were unable to sync Google Photos textures between them. A solution that we found for this issue was to have the client log in to Google Photos before joining the host. Utilizing this method, the client can view the photos without needing to be synced. Ideally, both the host and client would use the same Google account so that they can view the same photos.

### 3.5 Measurements of Success

#### 3.5.1 Quantitative Measurements

For quantitative measurements of success, our team has utilized a quality assurance (QA) process to validate the quality of our virtual reality application. The critical success metrics, which serve as key indicators for an immersive VR experience and profoundly impact a user's experience, are latency, frame rate, frame rate variance, and multiplayer synchronization. In our pursuit of delivering a unique, exceptional VR application, from week eight and afterwards, we have followed a continuous rigorous quality assurance process through each agile sprint to validate the quantitative measures of success.

Latency is the delay between the user's action and the application's reaction. For virtual reality, this delay needs to be very low in order to create an immersive environment. Hence, the goal is to reduce latency to ensure that every action within the virtual reality environment feels natural and is immediate. To address this, we have fine-tuned our app's frame rate and diligently worked on identifying (with the use of Unity Profiler) and minimizing CPU/GPU-bound bottlenecks, thereby reducing the delay and enhancing real-time responsiveness.

The frame rate is measured in FPS, indicating how many images are rendered and displayed on the screen in a second. Consequently, a higher FPS can result in a smoother, more responsive game experience (Jerald, 2016, p. 189)[[9]](https://www.google.com/books/edition/The_VR_Book/ZEBiDwAAQBAJ?hl=en&gbpv=1&dq=The+VR+Book:+Human-Centered+Design+for+Virtual+Reality+&pg=PR11&printsec=frontcover). The target overall frame rate is at least 90 FPS, as anything below this threshold will likely induce disorientation or nausea among users (IRISVR, 2023)[[1]](https://help.irisvr.com/hc/en-us/articles/215884547-The-Importance-of-Frame-Rates). Moreover, the frame rate is often adversely affected when the system is CPU/GPU-bound, leading to a less smooth and less responsive user experience. Our optimization strategies will target alleviating these bottlenecks to maintain a consistent and high frame rate, ensuring seamless and enjoyable user engagement. Optimization techniques such as code and content optimization will allow the processor to exert less effort to achieve a desirable frame rate (Umbra3D, 2018)[[2]](https://blog.hackerbay.io/why-optimization-is-everything-in-mobile-ar-and-vr-b54a5dc411b5).

On a side note, the level of frame rate we have measured highly depends on the type of device used. One set frame rate is not ideal for all devices as some, like a mobile phone, would have different power and resources to handle a certain level of FPS compared to a more powerful laptop. It is pertinent to note that while maintaining a high frame rate is essential, ensuring a low variance in that frame rate is equally or even more critical.

Variability in frame rates can significantly impact user experience, so other metrics to measure variability based on “The Effects of Frame Rate Variation on Game Player Quality of Experience” is the 95% frame rate floor metric (Liu et al., 2023)[[10]](https://dl.acm.org/doi/abs/10.1145/3544548.3580665?casa_token=ImXUD-jfVrkAAAAA:a1EScKaxyyGswiOaXWvQ_5XDJa4IjzTOjWpYQ6n2Y-CrCimkUTZUSnAAIKMpwaKqU7ajaQ_fahhc). This refers to the frame rate that is achieved or exceeded 95% of the time. So, if the 95% frame rate floor is 90 FPS, the game runs at 90 FPS or higher for 95% of the time. The paper discusses that users generally find the experience satisfactory, with good Quality of Experience (QoE), when the 95% frame rate floor is at least 80% of the target frame rate. Given this and that the consistent target is 90 FPS and that QoE is acceptable when a 95% frame rate floor is about 80% of the target frame rate, then our 95% frame rate floor would be 72 FPS. This means that the Visual Voyagers game application aims to achieve 72 FPS or higher 95% of the time the application runs to ensure an acceptable level of Quality of Experience for our users.

Lastly, multiplayer synchronization will be addressed with precision; as all previously mentioned metrics and network bandwidth affect this key application component. In a network, especially in a multiplayer application, bandwidth is a metric that limits the rate at which packets can be sent and received (Jerald, 2016, p. 420)[[9]](https://www.google.com/books/edition/The_VR_Book/ZEBiDwAAQBAJ?hl=en&gbpv=1&dq=The+VR+Book:+Human-Centered+Design+for+Virtual+Reality+&pg=PR11&printsec=frontcover). General techniques for reducing network traffic include data compression, selecting the transmitting of specific data, reducing resolution, and establishing audio peer-to-peer instead of through a server. Additionally, when many users communicate simultaneously, relying on a central server can cause bandwidth and processing bottlenecks, possibly leading to delays or degradation in audio quality. Peer-to-peer communication can mitigate some of these challenges by distributing the transmission load among the users themselves.

Through systematic analysis with Unity Profiler, iterative testing, and targeted optimizations, we have refined our virtual reality application to meet the established quantitative performance benchmarks throughout our software development cycle.

#### 

#### 3.5.2 Qualitative Measurements

For qualitative measurements of success, in order to meet the core objectives of Visual Voyagers, which creates a VR space where users can strengthen relationships with one another through viewing photos immersively, we have leveraged user feedback as a crucial tool. These metrics are essential for understanding our users' experiences, identifying emotional resonance, enhancing social relationships, and getting insight into the application’s usability and accessibility.

Therefore, engaging with our users through a tailored survey provides invaluable insights into the impact of our application on facilitating connections and evoking emotions or reminiscing memories.

The survey encompassed questions such as:

* Would you engage with this application again?
* How likely are you to recommend this virtual reality application to a friend or family member? If so, please explain.
* Did you feel empowered to share your thoughts or opinions while using the application?
* Have you formed or strengthened any relationships through interactions within the VR environment? If so, please explain.
* Were you reminded of a memory or event that you haven’t reflected on in a while? If so, please explain.

This feedback assesses the depth of emotional and social connections fostered by Visual Voyagers and pinpoints areas for improvement. Overall, there were 10 users surveyed, gaining approval from the majority. While most said that they would engage with the application again with their friends, we also received some criticisms.

#### 3.5.3 Validation Testing

To ensure a reliable user experience within our virtual reality application, we implemented iterative, unit, and end-to-end testing throughout the software development lifecycle. Every sprint saw new features and modifications undergo iterative testing to identify and rectify potential issues. Individual components or modules of our application will be subjected to unit tests, overseen by the respective team members in charge. As the software reached a level of maturity in its development, comprehensive end-to-end testing was executed, leveraging tools such as the Unity profiler to validate against our established quantitative success metrics and make adjustments as needed. An extensive list of question cases was considered during this testing.

##### 3.5.3.1 Question Cases

**User Cases**

* Did the user feel a genuine sense of “immersion” while in the virtual space? (i.e., did it feel “real?”)
* Did the user experience motion sickness or nausea while in the virtual reality environment?
* What was the most liked feature or aspect of the VR application by the user?
* Were there any unexpected issues or concerns raised by the user during their interaction with the application?
* Any other general comments, suggestions, or feedback from the user?

**System Cases**

* What is the minimum Internet speed necessary for users to load the application?
* How long does the application take to load on average?
* How would the user rate the responsiveness of the game controls on a scale of 1-10 and why?
* Did the user experience any lag or delays while interacting in the virtual reality environment with other players?
* Was the user able to easily communicate and interact with other players in the VR environment?
* Were latency issues encountered during the user's interaction with the application?
* Are users obtaining 72 FPS or higher for 95% of the time the application is run?
* Are users getting a consistently high frame rate while using the application?

**Edge Cases**

* What is the longest distance users can have from each other while using the app before latency becomes problematic?
* What happens if the user loses Internet connection in the middle of the app? How would nausea be treated in this case?
* What happens if there are more picture frames than the amount of photos a user has in their Google Photos?
* What would happen if the user deletes a photo from their Google Photos while the app is running?
* When multiple users are present in the environment, does the application maintain a performance of 72 FPS or higher for at least 95% of its runtime?
* How well does the application handle multiplayer synchronization? Were any delays or out-of-sync scenarios noticed?
* Are there any visible bugs or glitches when interacting with interactable objects within the VR application?

## 

## 4 Validation Testing Results & User Feedback

## 

### 4.1 Validation Test Results

After completing the program measurements were taken using the built-in Unity statistics tabs that recorded these measurements automatically. We found the FPS measurement to meet our constraint of 90 FPS exactly. The variance was also recorded, coming out to be 15 FPS approximately while the network packet loss was less than 5%. The ideal variance measurements should be less than 5 FPS as anything above 10-15 FPS is rather high while the ideal network packet loss percentage should remain between 1-2%.

Despite these other measurements being rather high, users wearing the VR headsets reported no real noticeable lag or synchronization issues when using the program in single-player mode, in multiplayer mode, or when using the Vivox voice chat feature.

However, the graphics of the game were laggy on laptop screens without dedicated GPU cards. On average there was 15-20 FPS when using these devices.

### 4.2 Evaluation Process

The process for user interviews was conducted in the same manner each time. The first usability of the product was tested by walking users through the different VR controls and logging into Google Photos for them. We allowed users to explore by walking through the outdoor and gallery scenes to immerse themselves.

A separate part of the process was held where we conducted a user survey after people were done using the product. We asked users for their own personal feedback that may have been missed during testing instead of asking them all the questions that we listed in section 3.5.3.1.

### 4.3 User Feedback

The following is an itemized list of the main feedback we received from our users:

* Nausea
* Glitching
* Avatars were hovering plane
* Teleportation for the client side was not as seamless as host
* Difficulty using the program that wasn’t calibrated for HTC Vive headset
  + Reticule was difficult on other non-HTC Vive headsets
  + Issues with keyboard
    - Issue with Relay code having to be in all caps (case sensitive)
    - No caps lock available
    - Just kind of slow in general

Some of the initial feedback that was received during the early user testing phases was that the spaces in the outdoor scene were too big, making it too slow and tedious to move around. The same can be said for the gallery scene, including that the walls were much too tall and the photos had awkward dimensions, stretching the photos oddly. We fixed this by resizing the environment and objects of the outdoor space and the gallery space to accommodate those needs.

Other initial feedback received was that users were not aware when they had logged into their Google Photos account or that there were no clear instructions on how to use the product or headset. This was accommodated by making the login button disappear after a successful login to a user’s Google Photos account and adding a guide button to the GUI, explaining how to play the game.

The most common feedback we received from users was that they experienced Nausea when wearing the headset and moving around. We do not believe that the slight lag was the main attributing factor to this but rather it was people’s general motion sickness from the close-up movement of the graphics.

There were also reports of difficulty using the program that wasn’t calibrated for the HTC Vive headset such as how when using the reticles on the controllers it was difficult for other non-HTC Vive headsets to click on the buttons in the GUI.

Some recurring issues with the keyboard were also a part of the initial and later user feedback. From the early feedback we received, users found the keyboard to be slow to use as they would have to constantly click on the shift key to use uppercase letters, and disappointment in no caps lock being present was also expressed. We fixed this by making the Relay input code to be not case-sensitive between upper or lowercase letters. However, during later testing, some reports that lowercase inputs for the code did not allow clients to connect to the host’s server. This issue was also seen during the live demo session as well.

Other feedback on minor issues that appeared during user testing and the live demo was some slight glitching, the avatars sometimes hovered above the plane, and teleportation for the client side did not appear as seamless as the host side.

### 4.4 Alternative Product Solutions

One of the earliest issues we faced when trying to test our product during development was that some of our devices did not have any dedicated GPUs or enough resources to use the VR headsets. We were concerned that future users would be limited depending on the specs of their devices as well.

However, during development, we came across an emulator feature that was made available under the XR Interaction Toolkit package on Unity (Unity Manual, 2023)[24]. This setting allowed us to develop and test our product without having to always connect our VR headsets or limit the number of people who could test the product. We then proposed creating a separate build of the product as a solution to the hardware limitations that the VR headset had. It was an emulator-specific build that didn’t require VR headsets. It could be used for emulator to emulator connection or for emulator to VR headset connection.

This also meant that the build with emulator capabilities would also allow for usage on other devices like Macbooks that were not originally compatible with our model of the HTC Vive.

## 5 User’s Manual

Visual Voyagers can be used in two different ways: using a VR headset (specifically the HTC Vive or Oculus Quest 2) or with an emulator. The VR headset is the best way to enjoy the application, however, we recognize that not everybody can afford a VR headset, which is why the emulator version is available. In the end, both versions of the app have the same features. After downloading the respective zip file and unzipping it, Visual Voyagers can be launched by clicking on the Visual Voyagers executable file within the unzipped folder.

When first launching the application, the user will be able to see various controls in the middle of the screen prompting the user to “Log In,” “Create,” and “Join” with the latter two controls pertaining specifically to the multiplayer functionality. “Log In” prompts the user to login to their Google Photos account. After pressing the button, the user will have to take off their headset (if using the VR version) to login through their desktop. Once they’ve completed the process, they will be prompted to return back to Unity, meaning that they can put their headset back on. Once they do so, they can host a room for other players, join someone else’s room, or just enjoy Visual Voyagers by themself.

On the left side of the screen, the user will be able to see a placeholder for the room code along with the instructions that describe the VR controls. The room code is created when the user is trying to host a room and the placeholder is replaced with an actual code after the user presses “Create.” Afterward, the user will be able to give that code out to whoever they want to join them in the app. If the user is trying to join someone else’s room, instead of pressing “Create,” they need to wait for the host to provide them with a room code. Afterward, the user types in the room code in the provided text box and presses “Join.” Immediately after successfully joining the host’s room, the client should be able to see and hear the host and any other clients around them.

### 5.1 VR User Manual

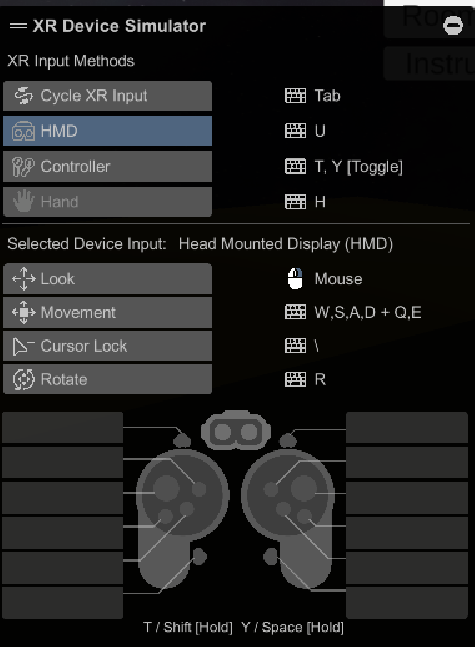
The compatible VR headsets for Visual Voyagers come with two controllers that allow the user to walk around and interact with the environment. Controllers typically come with at least a trackpad in the center and a trigger on the back.

The left controller has a teleportation ray and a circular reticle at the end of it. By moving the controller around, the user can control where they want to teleport, with the reticle demonstrating the exact location they will end up in. The left trackpad allows the user to move around. The left trigger controls the teleportation feature.

The right controller has an interaction ray that points out from the end of it. By moving that controller around, the user can control where they want to point and interact with. The user can only interact with buttons and the keyboard. The right trackpad allows the user to rotate. The right trigger controls the interaction feature.

### 5.2 Emulator User Manual

Instead of using a VR headset and controllers, the user can use an emulator on their PC or laptop. When opening up the emulator version, the following figure appears, describing how the VR controls map out on the keyboard and mouse.



**Figure 7.)** Keyboard and mouse instructions for the XR Device simulator.

## 6 Engineering Standards

### 6.1 Project Management

Progress was checked each week using the following Gantt chart and task log in order to make sure we stay on schedule with the project. In order to schedule and monitor progress, the Project Managers outlined a Gantt chart with the major tasks for the project, which was managed through Google Sheets. There was a shared Google Drive folder to share resources and notes, Discord to facilitate communication further, and Unity Version Control for development.

Furthermore, we had bi-weekly meetings scheduled during regular class time, 11:10 AM to 2:15 PM on Tuesdays and Thursdays. Any additional time needed outside of the specified hours was decided as a team on a per-need basis.

#### 6.1.1 Team Roles

**Estella Chen**

Responsibilities:

* Project Manager
* Frontend Developer/Environment Setup
* Found assets and created materials for scenes, created initial setup for hangout scene, worked on avatar functionality, added teleportation feature, added instructions, tested app using VR headset and emulator
* Conducted user testing

**Citlali Haro Franco**

Responsibilities:

* Fullstack Developer/Environment Setup & Multiplayer
* Set up Unity Version Control for project, found assets for avatars, added functionality for avatar, fixed integration for multiplayer synchronization between client and host

**Nick Nguyen**

Responsibilities:

* Backend Developer/Google Photos Functionality
* Set up Google Cloud Platform connection for OAuth and Google Photos APIs, developed and tested code to connect project to Google Photos, set up Vivox functionality, tested app using VR headset

**Annie Ren**

Responsibilities:

* Backend Developer/Multiplayer Functionality
* Set up Netcode multiplayer, developed and tested code for multiplayer functionality through Netcode, implemented Unity Relay, tested app using VR emulator

**Vivian Vo**

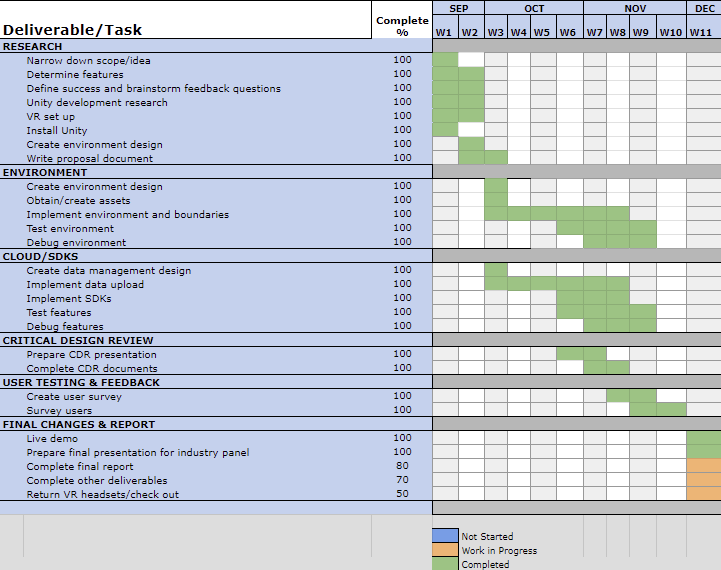
Responsibilities:

* Project Manager
* Frontend Developer/Environment Setup
* Found assets and created materials for scenes, created initial setup for hangout and photo gallery scenes, tested app using VR headset and emulator, assisted in implementing NetCode for GameObjects
* Conducted user testing

There has been one major modification to the team roles and responsibilities among the team. While there are still the two major roles of Frontend Developer/Environment Setup and Backend Developer/Cloud Setup, after starting development, Estella and Annie switched their primary roles due to the intensive setup requirements of multiplayer. By reallocating this division of labor, we were able to remain on track for the project and implement multiplayer functionality in time.

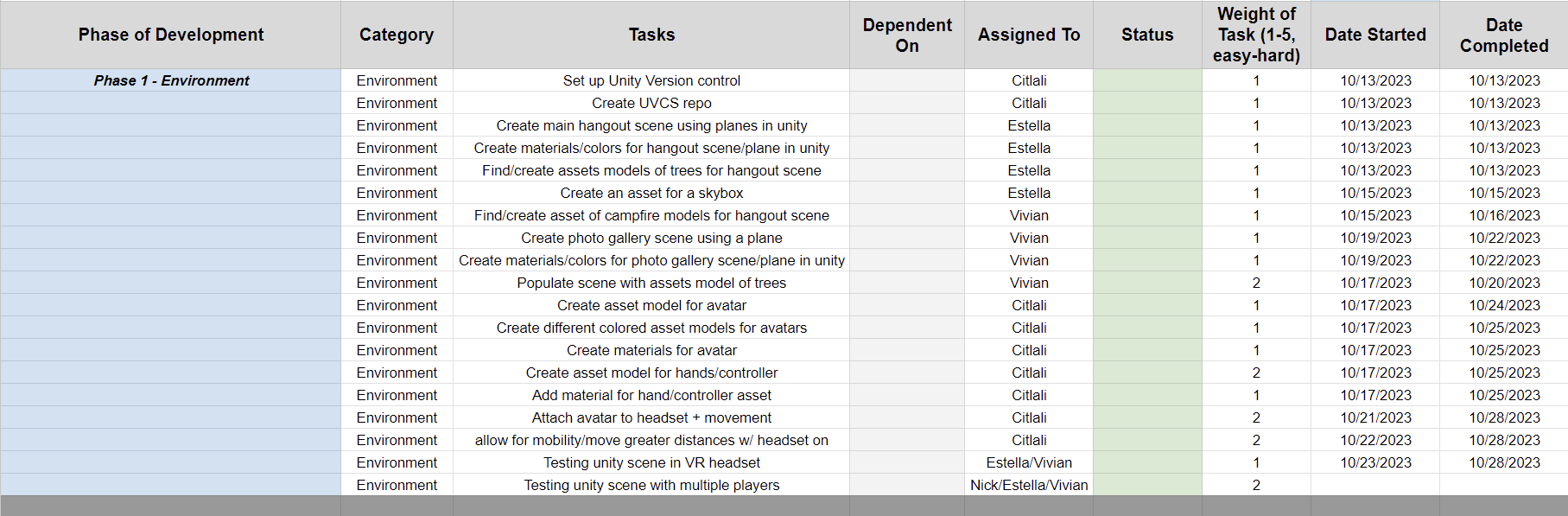
### 6.2 Updated Gantt Chart and Schedule of Tasks

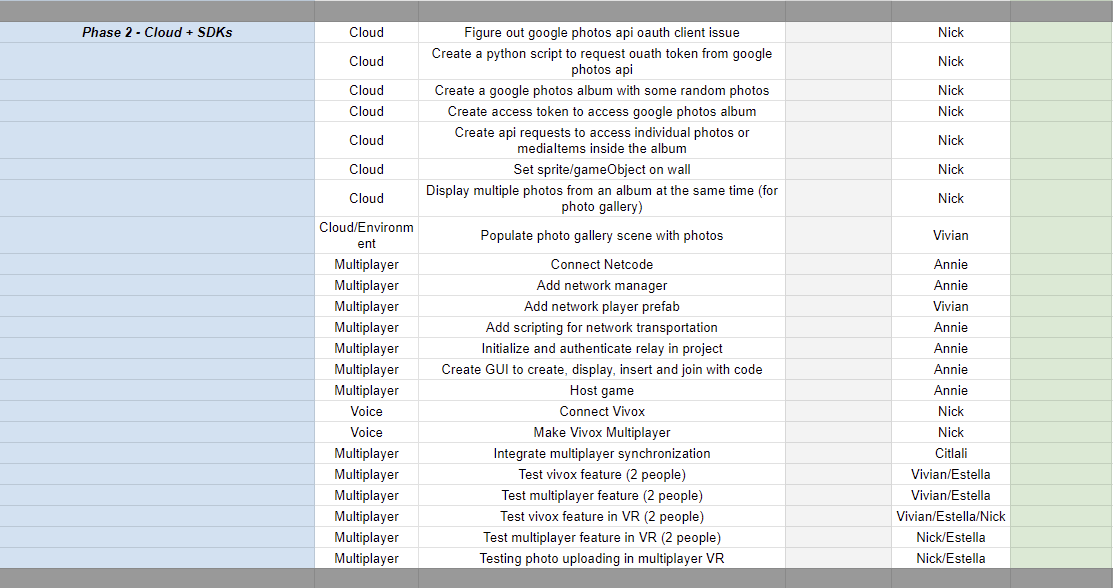
**Graphical view of Gantt Chart with Task Breakdown:**

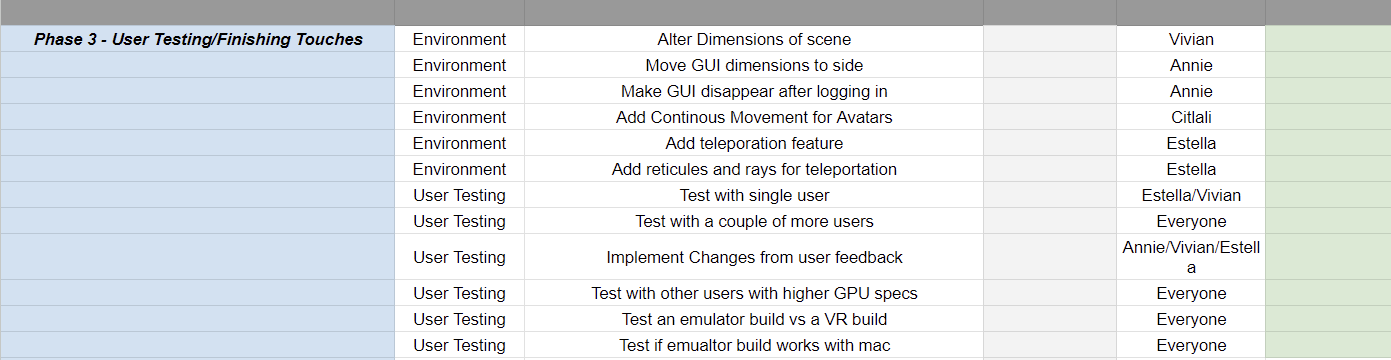


**Figure 8.)** Gantt Chart depicting the overarching goals or phases of the project and its subtasks.

**Task Log:**







**Figure 9.)** More specified breakdown of the major tasks and their subtasks.

### 6.3 Societal Impact

The societal impact of our virtual reality application, Visual Voyagers, extends beyond merely providing an interactive photo-sharing platform. By creating a space where individuals can reminisce and share memories in a more immersive and engaging manner, the application fosters deeper emotional connections and understanding among its users. As people navigate through the modern-day challenges of physical distances and social disconnect, Visual Voyagers provide a meaningful avenue to bridge these gaps, enabling the cultivation of relationships bound by shared experiences and stories. Over time, as users engage and interact within this nurturing environment, the ripple effect of these enriched social connections could positively promote understanding and shared human experience, bridging the gaps that physical distance may have created in strong relationships. Through the lens of virtual reality, Visual Voyagers is poised to redefine how memories are revisited and shared and contribute meaningfully towards building a more empathetic and connected society.

To provide an example, the societal impact of our application is particularly pronounced in the context of the elderly population, where social engagement holds substantial weight in enhancing life quality and longevity. For instance, research exemplified by Steinbach's study reveals that elderly individuals engaging in social activities nearly halve their risk of institutionalization while ameliorating the adversities of living alone, which, conversely, increases such risks (Steinbach, 1992)[[3]](https://doi.org/10.1093/geronj/47.4.s183). Participation in social engagements, alongside interactions with friends or relatives, has been directly correlated to a lowered likelihood of mortality.

Our application caters to this community, creating a welcoming space for elders to maintain social engagement with their family members, regardless of geographical distances. The ambition behind our application goes beyond mere online interactions; it stretches towards nurturing a sense of belonging and strengthening relationships. This project presents individuals living alone with compelling incentives for a fulfilling and enriched life, carrying the promise of enhancing their overall quality of life.

Drawing inspiration from a notable Harvard study, as cited in the book "The Good Life: Lessons From the World's Longest Scientific Study of Happiness," the indispensable value of human connections becomes glaringly apparent. The study articulates that individuals intricately connected to family, friends, and community tend to manifest higher levels of happiness and physical health compared to those who find themselves in more isolated circumstances (Waldinger, 2023)[4]. The repercussions of social isolation manifest not just emotionally but tangibly, as individuals who experience a sense of disconnection often encounter health decline faster than those with robust social networks. Moreover, a stark correlation between loneliness and reduced lifespan is identified. This grim reality is not localized but is an escalating global concern. It's alarming that about one in four Americans, accounting for over sixty million people, report experiencing loneliness (Waldinger, 2023)[4]. In this light, our virtual reality application aims to counteract this rising tide of social isolation by facilitating meaningful virtual interactions that strengthen the richness of relationships. Through our platform, we aim to fortify connections among individuals, which, as a collateral benefit, cultivates a space that can markedly enhance the quality of life, mental well-being, and overall longevity of our users by transcending physical limitations.

### 6.4 Ethical Impact

Due to Visual Voyagers being a virtual reality application, there are many ethical questions and concerns that may be raised. Despite how immersive and groundbreaking the simulated environments can be, this technology has the potential to both enhance and disrupt aspects of our lives, allowing the rise of ethical discussions about issues such as privacy and data collection, confusion between the real and virtual world, physical harm, among other concerns. Without keeping these concerns in mind, Visual Voyagers may accidentally cause its users harm.

One of the primary ethical challenges with VR is the amount of personal data that is collected and used in order to create these immersive experiences. VR platforms must collect an extensive amount of data on their users in order to correctly track their movements, speech, environment, biometrics, etc. while using the VR headset (Shion and Neelam)[[6]](http://www.cisin.com/coffee-break/technology/9-ethical-issues-with-vr-we-need-to-fix.html). Additionally, Visual Voyagers populates the photo gallery scene with photos from Google Photos, so the user’s data are shared with third parties. There are still risks associated with unauthorized parties gaining access to the user data; however, the data access is limited and requires user consent and Google authentication before continuing. Furthermore, the local aspect of Visual Voyagers will allow exclusively for smaller communities to engage on the application together, making it more difficult for hackers to gain access to individual users’ environments and disrupt their experience.

Another major concern of VR is the potential for confusion between reality and the virtual experience. Prolonged exposure to virtual reality can cause disorders such as addiction, depression, and loss of identity. This can be especially damaging to the vulnerable minds of children (Kenwright, 2019)[[5]](http://technologyandsociety.org/virtual-reality-ethical-challenges-and-dangers/). While this effect is not common and only happens in the most extreme cases, there is still risk involved for users. VR also seems to be especially harmful in the case of children, with studies showing that VR can enhance the lives of older adults, especially those in long-term care homes (Chaze, 2022)[[14]](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8832624/). Therefore, younger users who may be more susceptible to the immersion of VR must be watched by a guardian or parental figure during usage. Due to the emphasis on social interaction with friends or family during the experience, users will be able to easily keep each other’s screen time in check, encouraging each other to take breaks and step away from the screen.

With a VR headset, one’s vision is limited and they can only see within the virtual world. Therefore, there are concerns about physical damage while immersing oneself (Shion and Neelam)[[6]](http://www.cisin.com/coffee-break/technology/9-ethical-issues-with-vr-we-need-to-fix.html). While setting up one’s VR environment, there are different procedures in place for users who would like to sit or stand in one place versus users who would like to move around a certain space. This process is extensive and ensures that the user is in a big enough area for their preferred experience, reducing physical harm. However, Visual Voyagers is unable to track individual users and their choice of physical environment during app usage, so users themselves must ensure that their play area is clear of obstacles, potential collisions, and tripping hazards. Overall, there will be warnings and set-up instructions before users are allowed to step foot into the app, but it is up to individual users to clear their play area.

Overall, there are many ethical impacts of VR. Despite this, Visual Voyagers is taking these challenges in mind, ensuring that there are precautions in place and that users have the best experience possible. However, in the case of greater security concerns, responsibility should be maintained by the user.

### 6.5 Itemized Budget

**Hardware**:

* 2 HTC Vive VR Headsets ($0 since we rented them from TAMU Computer Science & Engineering Department)
* 2 Locks ($10 per item)

**Software**:

* Google Cloud Platform ($0 during 90-day free trial or before spending given $300 credit)
* Unity ($0 using a student/personal plan)
* Unity Version Control ($7 per extra member–first 3 members are free)

In total, this project has cost $34 since most items were rented or used a free account.

However, if this project were to continue, we would have to pay upwards of $800 for the same HTC Vive VR Headsets–upwards of $2,000 per headset if we wanted more updated VR headsets, a varying amount for Google Cloud Platform developer access, and upwards of $4,000 per year for Unity development.

## 6 References

[1] IRISVR. “The Importance of Frame Rates.” *IrisVR*, help.irisvr.com/hc/en-us/articles/215884547-The-Importance-of-Frame-Rates.

[2] Umbra3D. “Why Optimization Is Everything in Mobile AR and VR.” *Medium*, HackerBay, 30 Jan. 2018, blog.hackerbay.io/why-optimization-is-everything-in-mo

bile-ar-and-vr-b54a5dc411b5.

[3] Steinbach, Ulrike. “Social networks, institutionalization, and mortality among elderly people in the United States.” *Journal of Gerontology*, vol. 47, no. 4, 1 July 1992, https://doi.org/10.1093/geronj/47.4.s183.

[4] Waldinger, Robert J., and Marc Schulz. *The Good Life: Lessons from the World’s Longest Scientific Study of Happiness*. Simon & Schuster, 2023.

[5] Kenwright, Ben. “Virtual Reality: Ethical Challenges and Dangers.” *IEEE Technology and Society*, 14 Jan. 2019, technologyandsociety.org/virtual-reality-ethi

cal-challenges-and-dangers/.

[6] Shion and Neelam. “9 Ethical Issues with VR, We Need to Fix.” *Cisin.Com*, www.cisin.com/coffee-break/technology/9-ethical-issues-with-vr-we-need-to-fix.html.

[7] VRChat Legends Wiki. “Wiki Policy on Hacking.” *Fandom*, 20 Mar. 2021, https://vrchat-legends.fandom.com/wiki/Wiki\_Policy\_on\_Hacking.

[9] Jerald, J. (2016). The VR book: Human-centered design for virtual reality. (8th ed.) Morgan & Claypool.

[10] Liu, S., Kuwahara, A., Scovell, J. J., & Claypool, M. (2023, April). The Effects of Frame Rate Variation on Game Player Quality of Experience. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (pp. 1-10).

[11] Google. (2023). Google Photos. Retrieved November 6, 2023, from https://www.google.com/photos/about/

[12] SprocketVR. (2017). VR Photo Viewer. Retrieved November 6, 2023, from https://store.steampowered.com/app/531980/VR\_Photo\_Viewer/

[13] VRChat Inc. (2017). VRChat. Retrieved November 6, 2023, from https://hello.vrchat.com/

[14] Chaze, Ferzana, et al. “Virtual reality and well-being in older adults: Results from a pilot implementation of virtual reality in long-term care.” *Journal of Rehabilitation and Assistive Technologies Engineering*, vol. 9, 9 Feb. 2022, https://doi.org/10.1177/20556683211072384.

[15] Google. (2023, October 18). *Using OAuth 2.0 to Access Google APIs*. Google Identity https://developers.google.com/identity/protocols/oauth2

[16] Google. (2023, November 6). *Access media items | Google Photos APIs | Google for Developers*. Google Photos APIs. https://developers.google.com/photos/library/guides/access-media-items

[17] Unity. (2023, September 14). Vivox Unity SDK quickstart guide. Vivox Unity SDK documentation. https://docs.unity.com/ugs/en-us/manual/vivox-unity/manual/ Unity/vivox-unity-first-steps

[18] Unity. About Unity Transport: Unity Transport: 2.0.2. *Unity Transport | 2.0.2*, https://docs.unity3d.com/Packages/com.unity.transport@2.0/manual/index.html

[19] Unity. “Relay Servers.” Unity Documentation, https://docs.unity.com/ugs/en-us/manual/relay/manual/relay-servers

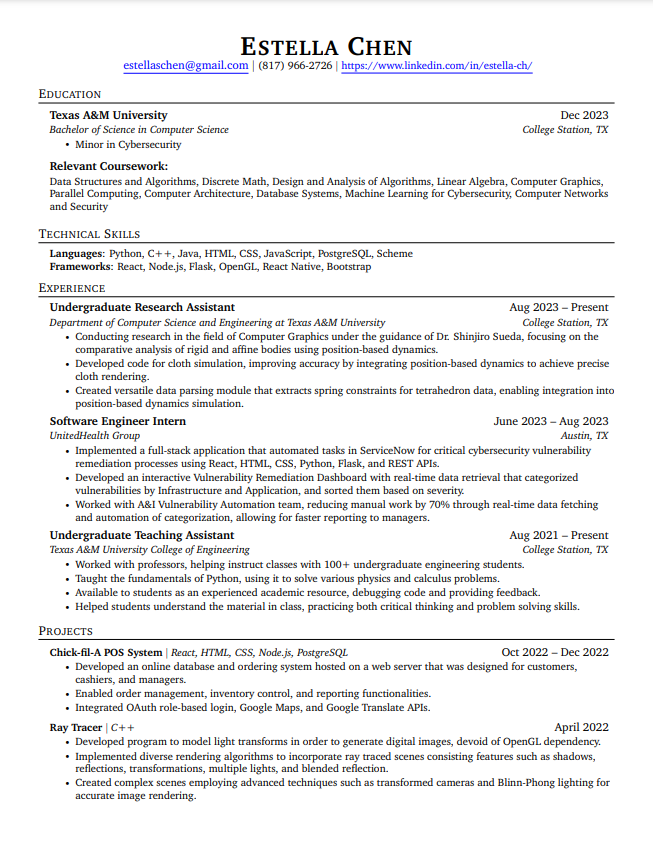
[20] Unity. “NetworkManager.” Unity Multiplayer Netcode Documentation, https://docs-multiplayer.unity3d.com/netcode/current/components/networkmanager

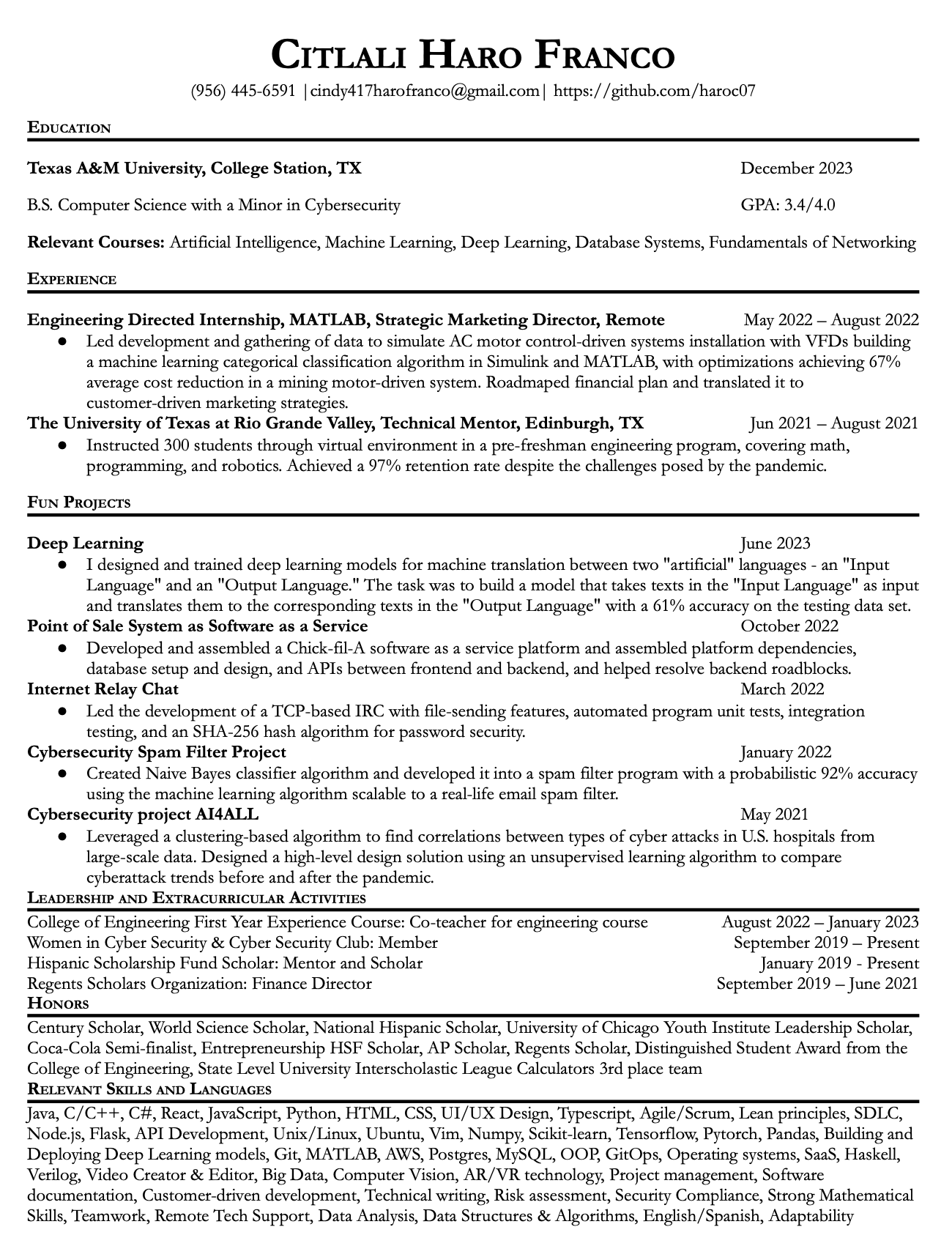
[21] Moody, Meaghan. “Oculus Quest 2 Specifications.” Studio X, 11 Sept. 2021, studiox.lib.rochester.edu/oculus-quest-2-specifications/.

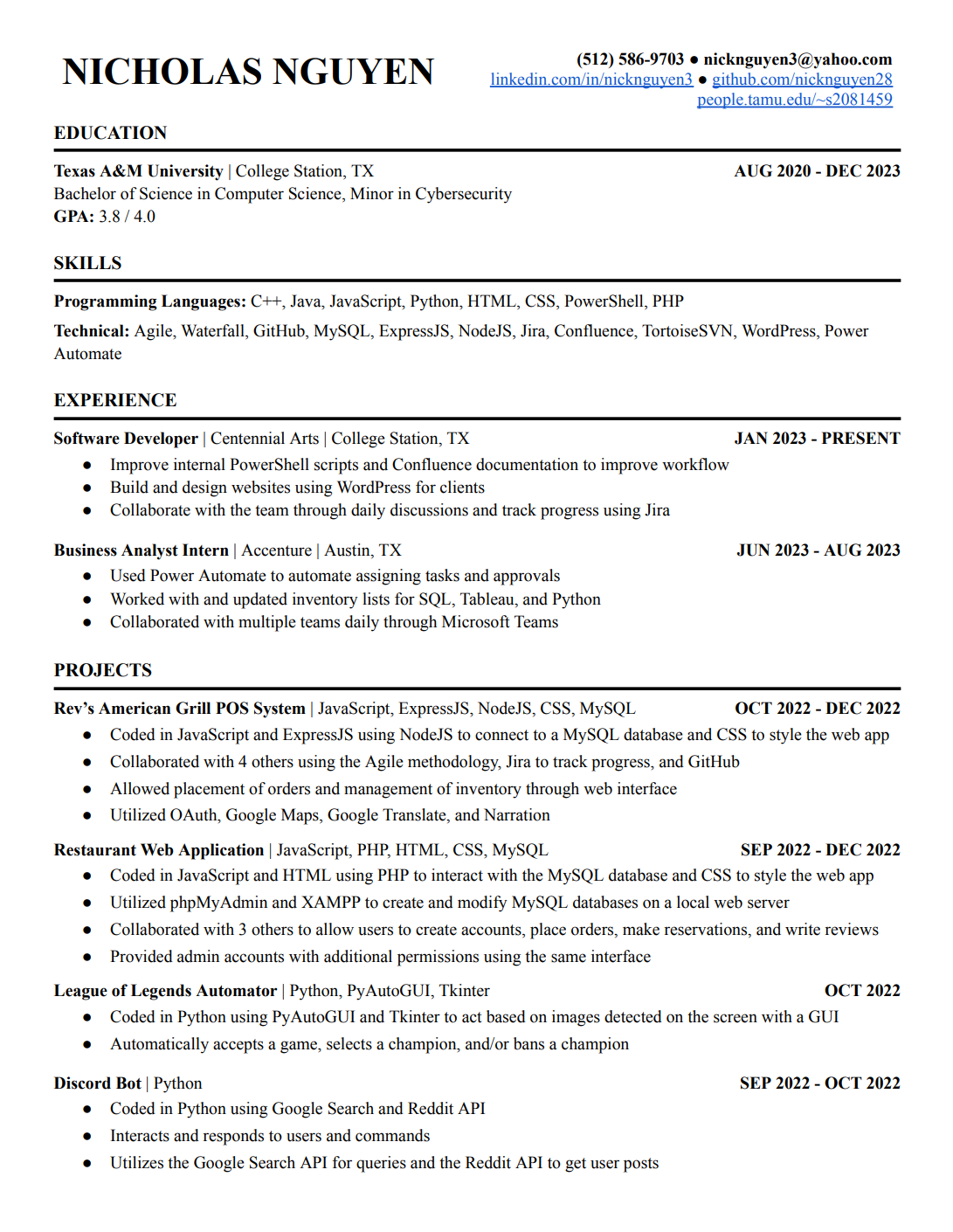
[22] Meta. “Requirements to Use Meta Quest Link.” Meta, 2023, www.meta.com/help/quest/articles/headsets-and-accessories/oculus-link/meta-quest-link-compatibility/.

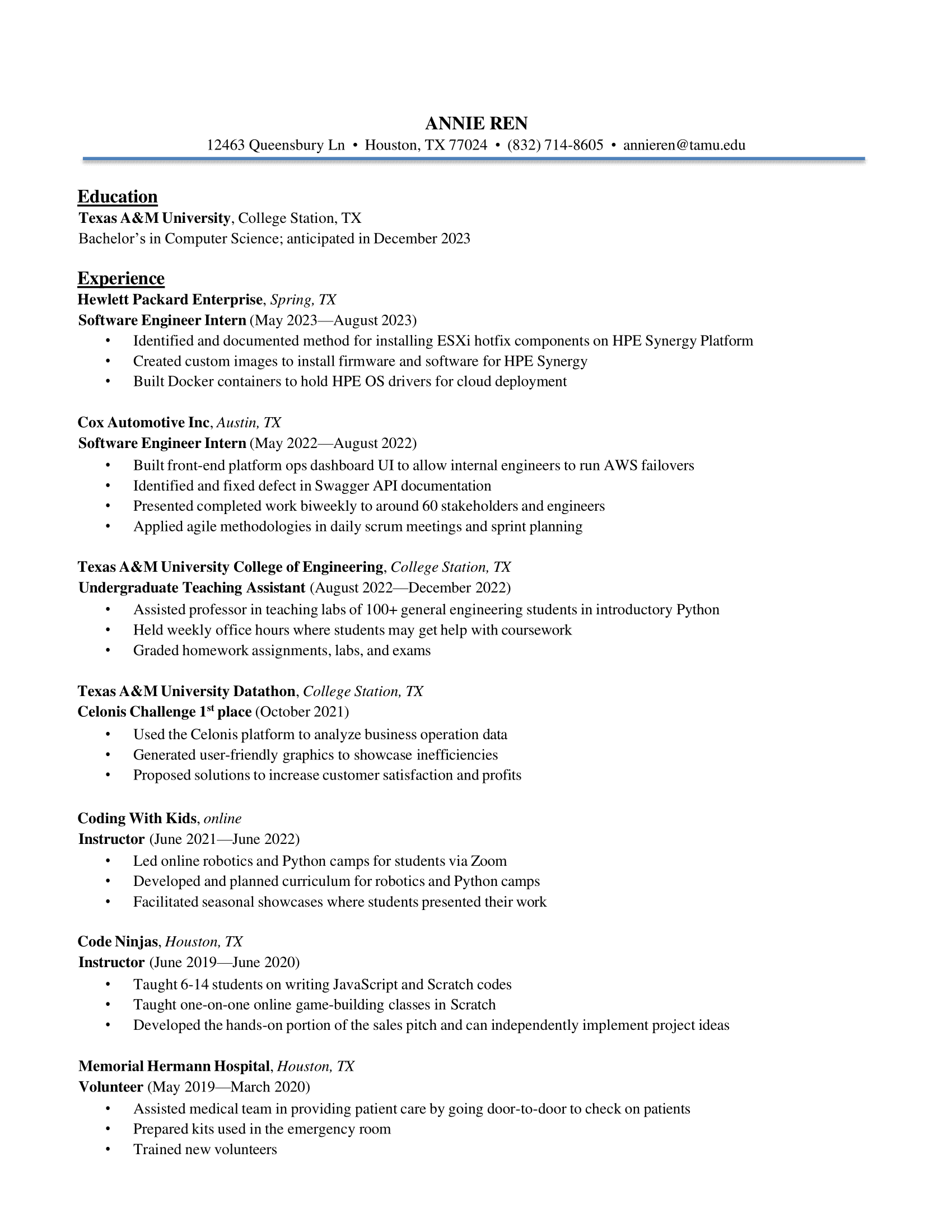
## 7 Appendices

### 7.1 Resumes









**Vivian Vo**

**817-937-8837 | voviv2000@gmail.com | linkedin.com/in/vivian-vo-400163219**

**EDUCATION**

**Texas A&M University,** *College of Computer Science Engineering (Bachelor of Science)*  ***\*Minor in Cybersecurity***

**College Station, TX** 08/2019 - ***Graduating*** 12/2023

**Relevant Courses**: Intro to Computer Systems, Computer Organization, Design Analysis Algorithms, Datathons, Cloud Computing and Amazon Web Services (AWS), Data Analytics for Cybersecurity & Machine Learning, Parallel Computing

**TECHNICAL SKILLS**

C++ | C | CSS | Java | JavaScript | GitHub/Git | Haskell | Hack Assembly | HTML | Linux | Microsoft Office (Excel, Word, PPTX) | React | Rust | Python | Jupyter Notebooks

**PROFESSIONAL EXPERIENCE**

**Hewlett Packard Enterprise (HPE)** - Software Engineering Intern *(Compute Unit - Enclave Team)*

**Spring, TX**  *06/2023 - 08/2023*

* Gained comprehensive experience throughout the development process, enhancing proficiency in various skills and aspects of the project.
* Attained proficiency in a new programming language (Rust) within 12 weeks, despite its learning curve through daily, self-directed learning and practice.
* Deepened my knowledge of hardware functionality and other low-level components like registers and boards.
* Developed Rust code to detect physical tampering instances while making use of triggering interrupts and sending debug messages as notifications.

**Undergraduate Teaching Assistant (TA),** *General Engineering*

**College Station, TX** *08/2022 - 05/2023*

* Assisted in general engineering courses, teaching coding concepts to engineering students using Python, resulting in enhanced coding skills for the students.
* Led 100% of working periods/labs and instructed classes on assignments, covering data structures and coding techniques, fostering a collaborative and engaging learning environment.
* Taught students debugging techniques and effective use of different data structures, aiding in their problem-solving and coding abilities.
* Graded assignments based on rubrics, providing constructive feedback, and encouraging students' learning and growth in coding proficiency

**Garmin** **Software -** Software Engineering Intern *(Core Technology - GPS/GNSS Team)*

**Olathe, KS** *05/2022 - 08/2022*

* Developed embedded software for Communication & Navigation products using C/C++, reducing errors, and accelerating production speed
* Produced software releases, and participated in product/code tests and maintenance processes, demonstrating effective teamwork in coordinating successful product launches.
* Conducted code reviews and collaborated with the other teams to deliver accurate and reliable software patches, ensuring high-quality products.
* Presented work at the intern project fair, receiving feedback and showcasing the impact of the project to the company, displaying excellent presentation and project management skills.