



**Department of Computer Science**

**BSCCS Final Year Project Report  
2022-2023**

**22CS014**

**VR Classroom- VR Application**

**(Volume 1 of 1 )**

Student Name : **BANBAH Kush**  
Student No. : **55786740**  
Programme Code : **BSCEGU4**  
Supervisor : **Dr LEUNG, Wing Ho  
Howard**  
1<sup>st</sup> Reader : **Dr ZHU Kening**  
2<sup>nd</sup> Reader : **Dr LAU, Rynson W H**

**For Official Use Only**

|  |
|--|
|  |
|  |
|  |
|  |

## Student Final Year Project Declaration

I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

VR Classroom- VR Application

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name: BANBAH Kush

Signature: Kush Banbah

Student ID: 55786740

Date: March 27, 2023

## Abstract

Online learning has become commonplace around the world on zoom or other video conferencing systems. This was a quick solution in a difficult time; it lacks interaction and must be improved. Virtual Reality is also rising in prominence and becoming more accessible, and no solution in VR exists for solving this problem of online learning.

This project will aim to solve this by creating a Virtual Reality solution in VR for Meta Quest 2. Students and teachers will join classrooms, communicate in a 3D environment and use an accompanying learning management system for students and teachers to upload course files. I aim to make this application more interactable and engaging and increase the quality of online education.

## Acknowledgments

I want to thank everyone who helped make my university life and this project successful. Thank you to the CS department and City University for giving me the opportunity and the tools to work on my dream project. My supervisor for supporting my project idea, offering suggestions and support, working flexibly with me and my group through the ups and downs of the project and being consistently available. Thank you to my parents for supporting me to attend university, their guidance, support through the past difficult four years. My sister for being a good role model and taking care of me throughout my degree. My grandparents for checking in and their wisdom and mental support. My cousins for being there to talk to and take my mind off matters. Lastly, I would like to thank my friends for their support, practical and mental throughout the project and university through the long and fun days, especially while working on this project with constant encouragement.

## Table of Contents

|  |    |
|--|----|
| Abstract.....  | 3  |
| Acknowledgments.....   | 4  |
| 1 Introduction.....  | 8  |
| 1.1 Current Solutions .....                                  | 8  |
| 1.2 Aims.....  | 9  |
| 2 Literature Review .....                                    | 9  |
| 2.1 Virtual Reality Solutions .....                          | 10 |
| 2.2 Online learning alternatives.....                        | 10 |
| 3 Solution Design.....                                       | 11 |
| 3.1 System Overview.....                                     | 12 |
| 3.2 System advantages.....                                   | 13 |
| 3.3 System Disadvantages.....                                | 14 |
| 3.4 Role of this FYP.....                                    | 14 |
| 3.5 Software and Hardware Selection .....                    | 15 |
| 3.5.1 Hardware Selection.....                                | 15 |
| 3.5.2 Software Selection VR Application .....                | 16 |
| 3.5.3 Software Selection Learning Management System.....     | 17 |
| 3.6 In depth Overview of the VR Application.....             | 18 |
| 3.7 In depth Overview of the Learning Management System..... | 18 |
| 4 Implementation .....                                       | 19 |
| 4.1 Design of the classroom .....                            | 19 |
| 4.2 Design of the characters .....                           | 20 |
| 4.2.1 Movement of characters .....                           | 21 |
| 4.3 Design of Whiteboard .....                               | 25 |
| 4.4 Design of UI and Networking Setup.....                   | 27 |
| 4.4.1 Virtual computer Interface .....                       | 28 |
| 4.5 Design of audio setup.....                               | 30 |
| 4.6 Quality of Life features.....                            | 30 |
| 4.7 Implementation of Learning Management System .....       | 31 |
| 4.7.1 Backend Services – Firebase .....                      | 35 |

|       |  |    |
|-------|--|----|
| 4.7.2 | Admin API – Java Spring.....                           | 37 |
| 4.7.3 | VR API – JavaScript Node .....                         | 39 |
| 4.7.4 | Learning Management Website .....                      | 40 |
| 4.7.5 | Security .....   | 41 |
| 5     | Showing files in VR.....                               | 42 |
| 5.1   | Creating textures at runtime.....                      | 42 |
| 5.2   | Sending the file to the VR headset.....                | 42 |
| 6     | Testing Plan.....                                      | 43 |
| 6.1   | Performance Testing of VR Application .....            | 43 |
| 6.2   | Performance Testing of Learning Management System..... | 51 |
| 6.2.1 | Admin API Testing.....                                 | 51 |
| 6.2.2 | VR API Testing .....                                   | 52 |
| 6.2.3 | Website testing.....                                   | 54 |
| 6.3   | Quality testing of VR Application .....                | 55 |
| 6.4   | Quality Testing of Learning Management System .....    | 56 |
| 6.4.1 | Admin API Testing.....                                 | 56 |
| 6.4.2 | VR API Testing .....                                   | 58 |
| 6.4.3 | Website Testing .....                                  | 60 |
| 7     | Final Results.....                                     | 62 |
| 7.1   | Results of VR Application .....                        | 62 |
| 7.2   | Results of the Learning Management System .....        | 62 |
| 8     | Conclusion .....                                       | 62 |
| 8.1   | Issues .....   | 62 |
| 8.2   | Analysis of the results and future extensions.....     | 65 |
| 8.2.1 | Learning Management System .....                       | 65 |
| 8.2.2 | VR Application.....                                    | 65 |
| 8.3   | Achievements.....                                      | 66 |
| 8.4   | Learning experiences .....                             | 67 |
|       | References .....                                       | 68 |
|       | Appendix: A Monthly Logs .....                         | 71 |
|       | A.1 March .....  | 71 |

|   |    |
|---|----|
| A.2 February .....                                  | 71 |
| A.3 January .....                                   | 71 |
| A.4 December .....                                  | 71 |
| A.5 November .....                                  | 71 |
| A.6 October.....                                    | 71 |
| Appendix: B Screenshots of Website.....             | 73 |
| Appendix: C Screenshots of Final VR Classroom ..... | 75 |

# 1 Introduction

Online learning became a familiar prospect in the last few years of the pandemic for many students and teachers worldwide. Many methods were used to solve this problem of creating an effective online learning platform, but all had some drawbacks (Wong, 2020). But before continuing, online learning should be clearly defined; online learning refers explicitly to learning when all learning is delivered through an online medium, and there is no face-to-face interaction. Learning refers to all educational-related activities that would happen in a classroom. For example, students ask questions, teachers give lectures, demonstrations, and other similar activities. While the pandemic may end, online learning offers significant benefits by removing distance barriers from learning and opening a wider world of possibilities. Students can learn from teachers worldwide and be exposed to new ideas (Miller, 2022). Online learning in this paper will not refer to students using an online learning management system like google classroom or canvas in conjunction with face-to-face classes. As an analog to in-person classes, a video conferencing tool like Zoom combined (Iqbal,2022) with a learning management system for students and teachers (Bouchrika,2022) is the online learning that is referred to, and this will be what is the target of this FYP. The current approach has a significant drawback regarding the interaction between students and teachers and its limitations. However, these benefits are too good to pass over, and online learning must be evolved.

Virtual Reality is a technology that can be useful in education. It provides a completely immersive environment, removes distractions, allows for communication, seeing other people, and more. VR market share has grown worldwide, in Statistica' (2022) analysis. As VR becomes more commonplace, and considering the benefits it offers, it makes it a possible technology to use for online learning to overcome its current issues.

## 1.1 Current Solutions

Online learning is widely popular and a growing market, but the current style of video conferencing as a medium to deliver the lesson needs fixing. The current interaction is based on being in a video conferencing room and allowing users to turn on their videos and microphones. But there is no need from the tool to keep the cameras on or speak, leading to

varied experiences, depending on the implementation of video conferencing in educational institutes. If the cameras are off, it leads to disengagement from the student and reduces the lesson effectiveness (Lemelin, 2021). Overall, there is a lack of relatability in learning when looking at classmates and teachers compared to physical classrooms for students, affecting education quality (Wong, 2020). This points to a clear area of improvement for online learning.

When it comes to Virtual Reality (VR) and education, it is very recent, and solutions still need to be created for the platform. The most common use of VR in learning has been pre-made scenes, lessons, or experiences like a medical student learning about the body. In a physical class, a teacher may give all students VR headsets and run a pre-made lesson. Hence, students would experience it as an addition to their lessons sometimes. Solutions such as Class VR (Class VR, 2017), Lenovo VR (Lenovo, 2020), and others offer such services but not a full-scale interactive classroom where real-time communication is possible.

## 1.2 Aims

This FYP aims to create a new system using VR and a learning-management system to overcome current solutions' drawbacks and offer real-time interaction in a simulated classroom. The students can wear headsets and experience an analog to a physical classroom. The system is designed to be a complete educational solution that can be built and used by an institute like a school or university for online learning. This report will explore the current implementations of online learning and VR in education and the design of a system that can solve this, it's testing, future features, and more.

## 2 Literature Review

A two-fold approach will be used to analyze studies to address current solutions. It will focus on the use of VR in learning and the current state of online learning. It must be noted that Virtual Reality is relatively new and has a few uses in education.

## 2.1 Virtual Reality Solutions

One large category of VR solutions in learning is video or lesson on-demand. These are offered by a few different businesses, such as ClassVR and Lenovo VR. Lenovo VR (2020) and Class VR (2017) aim to make VR a part of the lesson curriculum rather than using VR as a medium to have classrooms. They offer pre-made lesson plans and experiences such as virtual field trips, scientific experiments, examining engines, and dissection in biology. They also cover math, science, history, and more. These can be used for training professionals in their fields, like having mechanical students examine an engine piece by piece or having a medical student operate virtually. The opportunities and applications are incredibly vast and limitless. Each company offers lessons in all subjects, albeit each company does it differently.

There is one key difference between these systems and similar alternatives against the one this report describes. The others, as said before, are designed to add-on to a teaching experience. Lenovo VR (2020) and Class VR (2017) do not allow interaction between students and teachers or simulate a virtual classroom which is the aim of this paper. These alternatives may be robust additions to a lesson but are not intended to be used outside the physical classroom. They are designed to be stored together in carrying cases for charging, updates, and storage. Because their intention is different, these are not appropriate alternatives for replacing online learning solutions. They cannot be used separately and do not offer any method of communication between headsets or any virtual classroom.

Virtual classrooms and Metaverse is a concept that is quite new and solutions for this are being developed right now and, even in a year could be finished, such as Meta Horizon Workrooms. But at the time of writing, there is not a popular or fully ready application that is made for virtual classrooms and education.

## 2.2 Online learning alternatives

There are two main types of online learning methods: Video-On-Demand platforms and video conferencing tools. The primary replacement for physical classrooms that this paper is targeting is the video conferencing method. Video conferencing works better as a replacement for physical classrooms by scheduling meetings or conferences for students to join the class.

Video-On-Demand is often aimed at self-learning students to take courses in their free time and learn. Sometimes these can be a limited-time program where assignments have deadlines or be a course the student does at their own pace that they wish to do (Fordham University, 2020). This style existed before the popularity of video conferencing style and is not the target for this paper. Video conferencing allows as much interaction as possible with a live video chat where students and teachers can talk and see one another on the screen. However, this interaction is only sometimes present; students and teachers can switch off their videos. The students may get distracted during a lecture or explanation, and the teacher would need to realize the students are no longer paying attention. As Ramachandran (2021) and Shockley et al. (2021) discuss, with video calls, always having your camera and your face visible can sometimes get emotionally tiring; families could have their living rooms filmed while the student attends class there, privacy is a concern. Alternatively, if students turn off the camera, it makes it more isolating for the teacher. The students also disengage more from the lesson and become distracted (Lemelin, 2021).

The current solutions for online learning are limited to only video calls, and even that comes with issues. VR can overcome the problems better because it does not require a camera and its associated issues. Instead, it transports the user by creating a virtual environment where users can see each other walk around, observe some body language and limit distractions by changing the environment entirely. Newer headsets can track eyes, expressions, complete body posture and more but are not accessible yet. Current developed VR solutions are aimed at being extensions of a physical classroom rather than a medium. This leaves a gap for a VR solution aimed at solving the issue of online learning.

### 3 Solution Design

A possible product that can fill this gap would require combining the benefits of VR with online learning. In the current ecosystem of using a learning management system combined with video conferencing, a VR application would be needed to replace video conferencing. This VR Application would need to provide the same chatting capabilities, and some sort of substitute for screen-sharing and annotations since it is a well-used feature. A VR classroom

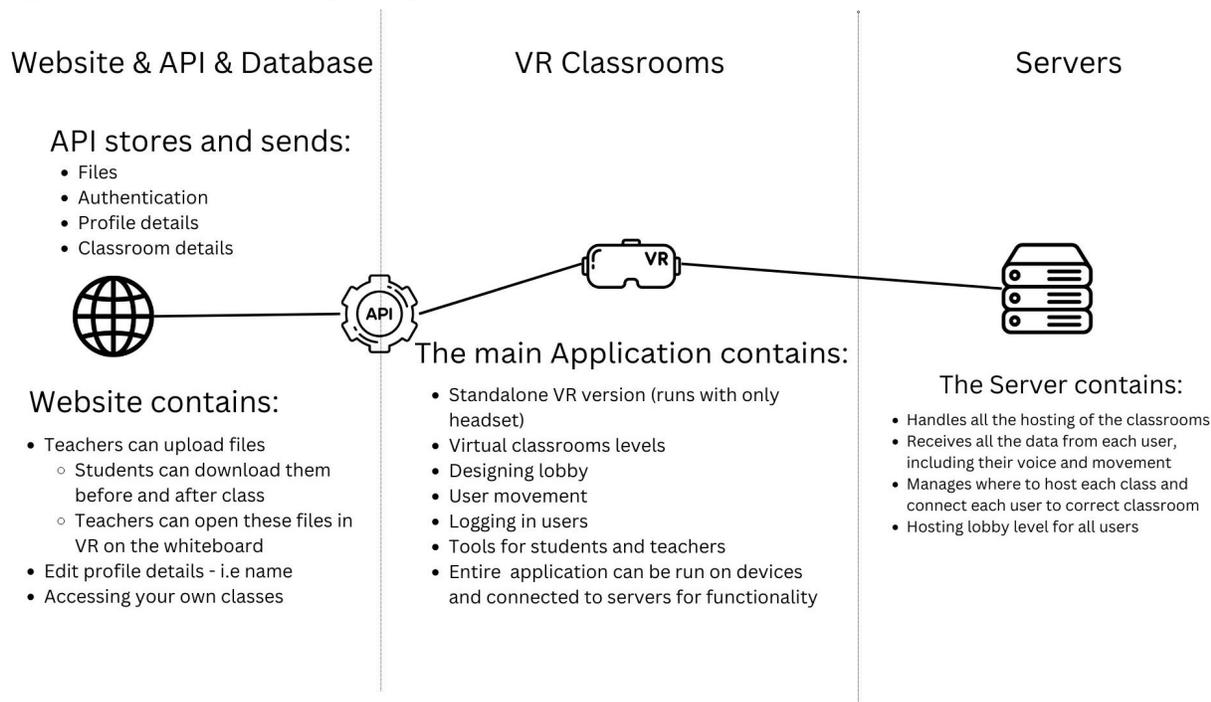
that could replace this would need to have a way to let teachers share lectures or presentations while they talk. There should also be a system to make uploading these lectures or presentations easy and quick. The application must be as simple as possible for teachers and students as an educational tool. As an immersive experience, it needs to simulate the real world; small touches such as shadows, character design, footsteps, and more would enhance building an accurate classroom simulation. However, this depends on technical limitations.

An accompanying learning management system would need to be designed that can, at the minimum, hold user information, files, and classroom information. Because of the tight integration required with the VR Application, the learning management system must be made in-house and not a third-party solution.

### 3.1 System Overview

The system would have three major components: a Learning Management System, a Virtual Reality Application, and the servers managing the classroom hosting. The system is designed for the stand-alone VR headset 'Meta Quest 2', so a computer is not necessary to join the classroom.

Figure 3-1. A system design diagram to explain breakdown of systems.



The Learning Management system will be a website that will allow teachers to upload documents and for students to download them. The Learning Management system (LMS) will consist of a full-stack website. This includes a front-end website, APIs to facilitate the VR Application and website, and finally a database to store all user information, files and class information. This LMS will make the entire solution an end-to-end system and not require any other products to accompany it for educational institutes. It also solves the problems of uploading documents, managing profiles, class information, and more for each student and teacher; these tasks can now be handled by the LMS for the VR application.

Lastly, online servers will be used to host the classroom sessions to which the users will connect. Teachers and Students will connect to a classroom, walk, and talk in the virtual environment. Teachers will be able to show pdfs on whiteboards, annotate, and more to keep the functionality of a classroom.

### 3.2 System advantages

The main criticism of interaction can be overcome by having the student be in close to real-life experience in VR. Instead of just looking at a flat screen, they can be in a classroom. They cannot get distracted as easily inside the headset since looking at phones and being distracted by their real-life surroundings is harder when they are not visible. It also helps with the privacy issues caused by having the camera on constantly in video conferencing (Ramachandran, 2021). Seeing a simulation of the teacher as they walk around, talk with students, and see more than just their faces like on video calls will lessen distractions. Looking around 360 degrees and all the other features will offer a much more immersive experience and come closer to a real classroom. It also offers easier multiple way interactions, students can talk to people around them, teachers can talk to all students at once or to some students. This is possible by using audio in a 3D space where only nearby people, like in real life, will hear each other talk. This makes it easy for groups to form in class, different types of interaction and more. Breakout rooms and other tools are not needed, it works naturally as it does in real life. So multiple people can talk to each other at the same time and be heard, or a single person can address multiple people and any combination of these.

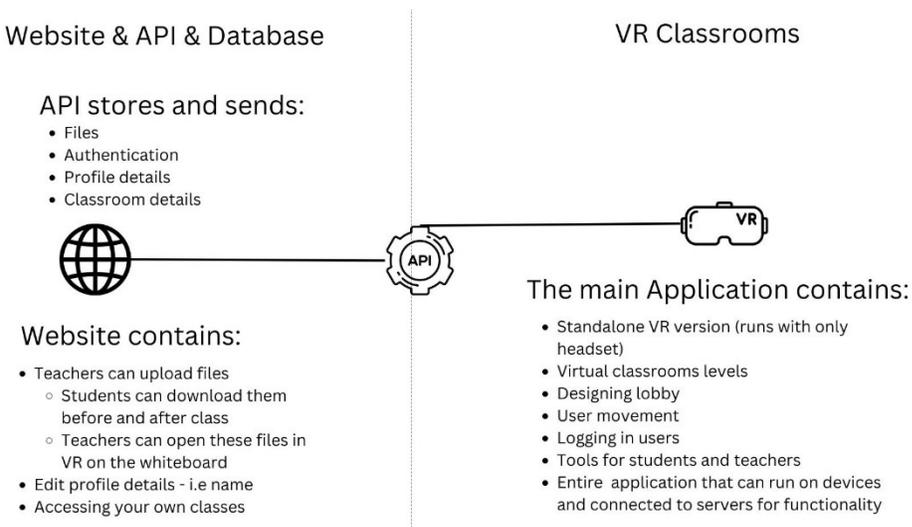
### 3.3 System Disadvantages

Some drawbacks must be considered for the VR system. Being immersed in a VR world for hours is much more exhausting on the body than watching a screen (Klemens, 2022). For students, especially younger children, this burnout must be monitored and is a concern. In the future, this should become a less serious concern with improved hardware and technology advancement. Compared to physical classrooms, students still need to work on assignments in VR and show their teachers as they might in a physical classroom. The curriculum and teaching style would have to be adjusted as well. These drawbacks in time with more software and hardware development can be overcome and alternative techniques such as using AR and VR together can be developed further and be viable. However, this solution will be more effective than current online learning.

### 3.4 Role of this FYP

This FYP is a group project, and the overall system is split between two people. This FYP is focused on developing the VR application and the accompanying learning management system. If it must be executed on the headset or if it concerns the backend services of the website or the website itself, it will fall under this FYP. To clarify, backend services will refer to the user authorization, database, file storage and other similar services to facilitate the learning management system and VR Application. The servers for the VR Application that will host lobbies and classes are handled by another group member, all the multiplayer sections and aspects of this project are not under this FYP. For this FYP, only the initial setup will be made for those servers, essentially creating end points for the server side to pick up. The entirety of the virtual classroom, the user features, tools, will all be the responsibility of this FYP on the VR side as well as designing the website, setting up the backend services, and related APIs for communication. Going forward in the report, the testing, implementation, and other sections will all refer to the content of this FYP (VR Application and LMS) unless system-wide is specified.

Figure 3-2. Sections covered by this FYP



### 3.5 Software and Hardware Selection

#### 3.5.1 Hardware Selection

To accomplish the goal of making the system as easy to access as possible, the main requirement was an all-in-one headset. The headset chosen for this was MetaQuest 2 (also known as Oculus Quest 2). There are two types of headsets, all-in-one headsets and pc-dependent headsets. PC dependent headsets require a separate computer to connect to and power them. The headsets use the computing power of the pc and display the application running from the computer.

Comparatively, all-in-one headsets are all encompassing, they contain the hardware and software to run supported applications. They do not require an external computer to function. This puts a limit on the computing power, since the best computers would outperform a mobile CPU like the Qualcomm Snapdragon XR2 in the Meta Quest 2. However, not everyone would have a computer powerful enough to support pc dependent headsets. And if they do, performance will vary depending on the spec of the computer, causing applications to run differently. This would affect the users by creating inconsistent experiences. However, this problem is surmountable by setting a minimum requirement and adding a system to balance the quality of the visuals against the performance of the application to account for different computers. But another issue with pc-dependent headsets is the requirement of always

connecting to a computer. This necessity would make it so users would require owning two different devices and more complications. The computer would also require a wired connection to the headset making it cumbersome as they walked, offering a worse experience.

All-In-One headsets offer a smoother process for development, a consistent experience for every user and are easy to use. Developing different all-in-one headsets at the same time is possible but having a targeted headset for creating the experience on is important before aiming to develop for more all-in-one headsets. As Meta Quest 2 is the most common (IDC, 2022) all-in-one headset that also offers controllers and enough power for the VR classrooms application, it is the targeted headset. Other options for all-in-one VR use phones for displays and are not suitable for the level of processing and the features planned for the VR classrooms application. At higher prices, there are alternatives such as Pico 4 and others.

However, Meta Quest 2 is the cheapest and an extremely popular headset (IDC, 2022). This makes finding support for development much easier and targets the most common headset. For future longevity, it is easy to upgrade and run the application on future Meta headsets. Meta Quest 2 offers a mix of accessibility, price, power, and ease of development, making it an ideal choice for this project.

### 3.5.2 Software Selection VR Application

There are two main choices of software for complex VR applications that require interactions. Unity or Unreal Engine. These game engines support the entire application, allow users to develop and deploy to devices with ease, and both support VR. Without game engines, development is nearly impossible for applications because of the framework and support they offer. Unreal Engine and Unity are widely used in the industry; *Beatsaber*, a VR game that sold over a million copies (Matney, 2019), was developed on Unity and won the Unity award of Best VR game 2018 (Unity Awards, 2018). *Vader Immortal*, an officially licensed Star Wars game, was created using Unreal Engine (Epic Games, 2019). Both engines support multiple platforms, from VR to PC to console development. Unreal Engine has the support of Epic Games who is well off financially (Browne, 2021) as its developer and is constantly being invested to gain developer share. Unreal Engine 5 recently released and boasted features such as nanite and lumen which will be implemented on VR in time and offer instant benefits in terms of performance and

quality of visuals, while Unity is working on counterparts to these technologies that will release in time.

Unity and Unreal Engine both are ideal and have no strong disadvantage or advantage over each other, hence the more suitable engine is based on personal preference. Based on developer choice, Unreal Engine 5 was chosen for its language of C++ over C# and personal ease of use.

### 3.5.3 Software Selection Learning Management System

The learning management system will consist of a front-end website, the backend services, and an admin API for communication with the website and the VR Application. As the focus of the FYP is primarily the VR Application, the learning management system must be simple and functional. To keep it simple, instead of spreading the services such as database and authorization with different systems or frameworks, an all-in-one backend service will be used. This saves in development time and these services are trusted, used globally, and backed by large companies such as Google, Amazon, Microsoft. The services may reduce the level of customizability and offer ease of use, with the simple requirements of the project this is preferred. The backend service provider will be the base of the entire project, it should be chosen first and decide the rest of the choices for API and website. There are a variety of cloud providers who will provide file storage, authentication, and database services. There is Firebase, AWS, Azure and Google Cloud Platform, any of these services would be able to provide all the necessary features, due to more personal development experience, Firebase will be used as the backend service. It offers simple native SDKs in JavaScript, Java and other various languages to make development straightforward and easy. It is also scalable in future and generally easy to work with, especially since the project is not complex, hence making setting up basic services easy. Since both website and the VR application will be requesting similar resources from database, having a single API that can receive these requests and forward them to database will streamline and standardize the process of accessing the backend for different platforms and add a layer of security with having the API verify requests. The API will manage all the internal communication with backend instead of direct access. There are again many choices for how to develop an API with firebase, there are certain languages where firebase admin SDK is

supported and due to experience with language and framework, Java Springboot is chosen as the framework. The API can be hosted on any Spring API hosting service such as fly.io with HTTPS. The website should be simple and functional so a straightforward JavaScript, HTML, CSS implementation will also be sufficient rather than overcomplicating with other front-end frameworks.

### 3.6 In depth Overview of the VR Application

The VR Application must provide an immersive and responsive experience for the user and provide features to facilitate the virtual classroom. To accomplish this, VR application development is split into design features such as creating the classroom and model design, and other features designed to improve immersive and responsive experience. The second part of the VR application is focused on the tools provided for the classroom. Such as creating a whiteboard, downloading pdfs from the Learning Management System, and displaying them on the board for the teacher to annotate and scroll with. These features will all consist of challenges discussed in implementation. Furthermore, exporting to the headset requires balancing out rendering techniques, visuals and hardware limitations will be an important concern and discussed in the testing section.

### 3.7 In depth Overview of the Learning Management System

The system must provide a functional website with authentication and authorization. Teachers must be authorized to upload files to their registered classes and authorized users can view files for download. There must be an API to provide user information to the website and VR Application such as what courses they belong to and what is the user's name. Working on web development has unique challenges of accessibility, performance, and network issues. The system must quickly be able to do basic actions like login, store files and user data securely and have strong error handling for network issues. Managing security of HTTPS and making sure the data is secure is another factor. Network issues and other errors such as lack of authorization, incorrect credentials, missing files and other small issues could cause a program to fail. With multiple co-existing APIs and services, this needs to be monitored to create a robust system.

## 4 Implementation

### 4.1 Design of the classroom

Figure 4-1(a) and Figure 4-1(b). Early screenshots of the classroom taken in unreal engine during development and do not represent final look in VR



The design of the classroom is based on a CityU lecture hall. It has four rows of students with tables, each on a different stair level with a large 'ground' floor where the teacher will be. The wall features a whiteboard that will also display the teacher's selected pdf from the

learning management system. The students have the option to sit in their chairs or walk around the class.

A challenge in the classroom is the model and material design. Materials in unreal engine are made of textures which describe what should be the color of a mesh or model. For example, to create a specific color, a painter would mix two different paints (textures) in a certain way and use special techniques to create a new paint (material) to apply onto a wall (mesh/model). Materials can be instanced as well with parameters to allow easier modification and change at runtime. Materials and models need to be within certain complexity to attempt to be realistic while ensuring performance does not suffer. The model cannot contain too many details or polygons, and materials must not be overly complicated with expensive functions or detailed textures. This consideration will be important while developing the classroom design to prevent stuttering performance from causing negative impact to the user. If the headset does not run at the specification consistently, it can cause headaches, dizziness and more (IrisVR, n.d). Rendering and performance related challenges will be discussed in the Testing section to prevent framerate issues.

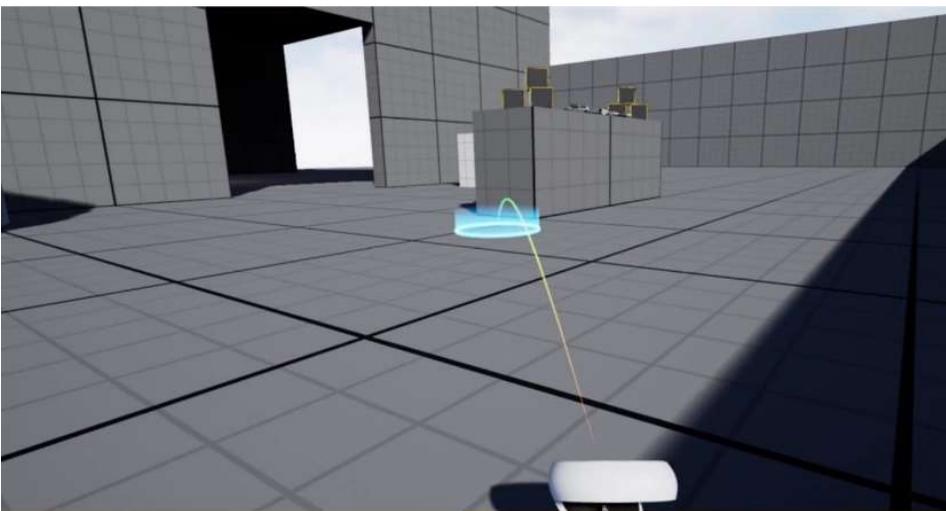
#### 4.2 Design of the characters

Before continuing, it is important to clarify what a character is. It is the representation of each user in the VR classroom. It is a class containing the visual and functionality logic of a user, for example describing how they look, how they move and act like. Each user needs to have their own 3D model that will move according to their body. The character's head would follow the user and its arms would follow the user's as well. The avatar must follow the user's movements and will require some sort of system to accommodate this. The arm and body pose of the avatar are an estimation from using the user's hand locations and orientations combined with the head's orientation and location. These can all be tracked and used to estimate the rest of the pose via a third-party plugin inspired on the Inverse Kinematics work of Prager (2018). This is all used with the default Unreal Engine avatar that will represent each user.

#### 4.2.1 Movement of characters

For movement, there are various types of VR locomotion . For this project, 3 styles are adopted; continuous and non-continuous artificial movement and motion based physical movement. Artificial movement refers to any movement that relies on controllers and is not naturally moving. Physical movement here refers to physically moving in the real world to move virtually. (Boletsis & Chasanidou, 2022) VR locomotion has many different implementations, and for this application it must suit a wide variety of scenarios. If users do not have large amounts of space, artificial movement is necessary. If they do have space, they should be allowed to take advantage of it and use physical and artificial movement types. For continuous movement, a smooth movement style with controller will be used. For non-continuous a teleportation system will be created. The smooth style is more jarring for newer users but may be preferred by more experienced users. This is because smoothly moving while physically being in place can cause motion sickness without previous experience in VR. Teleportation creates instant movement but combined with a fade to black, it smoothly repositions users and is easier for them to adjust to. For users with large spaces, they can walk in whatever space they have, and if they wish to go beyond, use artificial movement types to compensate. To accommodate variety of users, implementing user-friendly default options and giving more advanced users options for movement is the approach.

Figure 4-2. Sample implementation of teleporting style movement. Picture taken from implementation in Unreal Engine default VR starting level.



Regarding the teleporting style, when user's push their joystick, the application displays a line moving in an arc pointing to a location where the user can teleport. To achieve this, navigation tools of unreal engine and vector calculation were used.

Figure 4-3(a) and Figure 4-3(b). Screenshot showing which areas can be navigated or teleported to via teleport movement and screenshot of movement implementation demo.



In the figure above, navigation was set up and is represented by the green areas. The computer area on the right side is not teleportable and the main student area is not either. The chairs are interactable and can be sat on but not teleported to prevent people teleporting in a small area. The linear algebra algorithm casts a projectile arc from the user's right hand's forward vector, which is the forward direction. The projectile function returns an array of

vectors along the path of the projectile, at these points a spline mesh can be generated to create a line for the user to visually see where the arc is landing. The end of the arc is then checked across the navigable areas, and if the arc does not end in a valid area, then the closest point to the arc's landing location within a pre-set area is used. If a valid point is found, then the teleport location X, Y is established with offset of the user's camera, which represents where the user is in real life from the teleport location. Without the offset, if the user moved around the room physically, the offset of their movement in real life would be how far away they teleport from the projected teleport location. For the Z value, this is not an issue since the user is not going up or down a staircase or any large incline.

The alternative smoother movement option also exists. This design is based off standard movement style of first-person games using a controller. One of the user's joysticks is used to look around and the other will be used for movement in any 4 directions. The left joystick will be used to adjust location and the right joystick to adjust rotation of the character. The movement requires calculation to know which way to move. The in-game camera is the representation of the user's point-of-view (POV). The character will determine the forward direction of the user's camera and then move in the direction of where the user is looking. So, if a user looks to the left with their headset, and moves their joystick forward, their character will turn to match the direction, then begin moving left. This is because in real life, humans do not have to think which they are going and are looking consciously; hence users will move in the direction they look at. Alternatively, another implementation could be the user's character would not turn to match the direction they are looking at, and they would walk in direction with respect to the character instead of the direction they are looking at. However, this implementation feels unnatural from testing so is not implemented.

This is done by retrieving the forward vector of the in-game camera. The forward vector indicates the forward direction, then multiplied by a scalar to give movement input to the engine. The scalar is the value of the thumb stick input, a one for forward, negative one for backward. For left and right movement, the same approach is used, but using the right vector instead of forward vector and the scalar will use the thumb stick turning right or left. Using these systems simultaneously creates the entire movement system in any or all four directions.

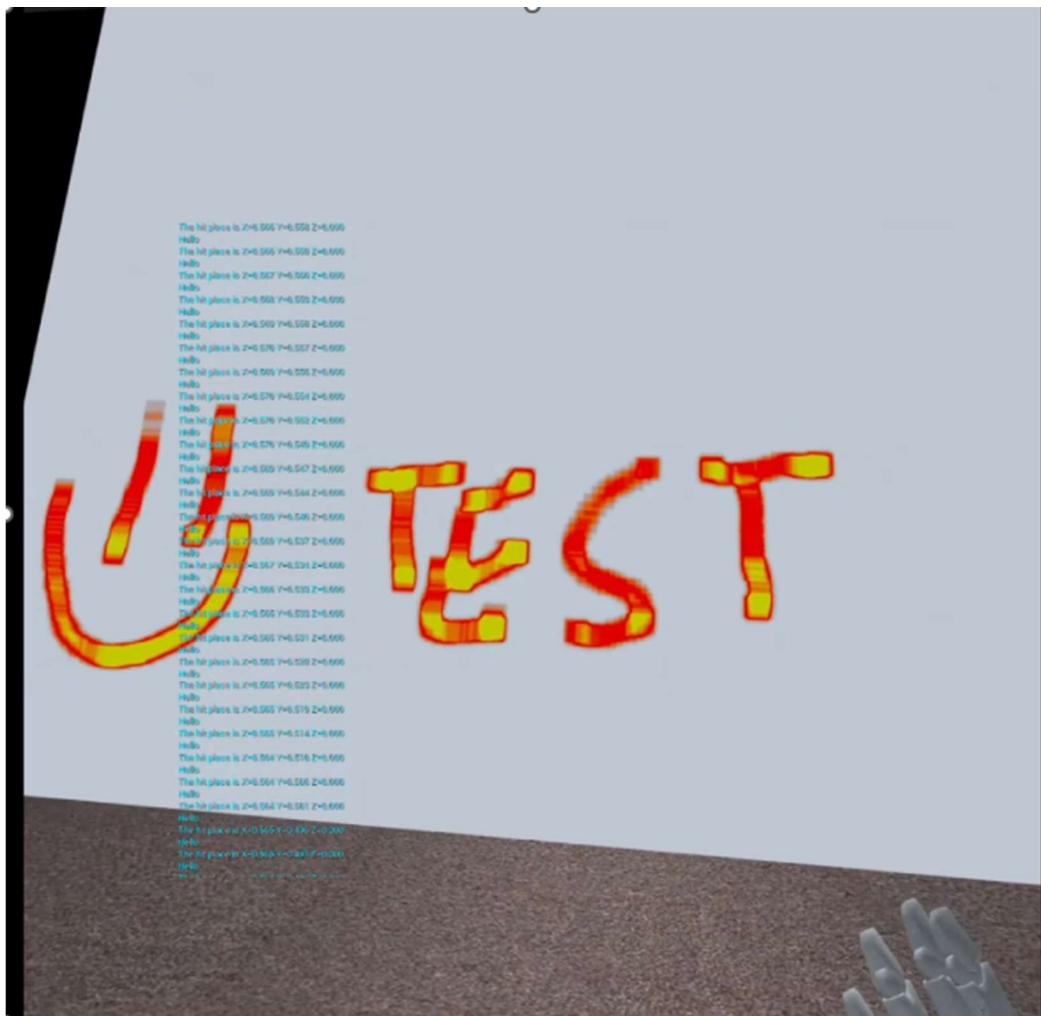
For rotation, the right joystick is used to generate a scalar value of one or negative one based on the thumb stick input of left and right. The current rotation of the character is retrieved, then the character is turned to the desired rotation. Alternatively, snap turn is another approach to the smooth-turn method, where the user turns a preset number of degrees instantly when moving the thumb stick left or right. This is implemented with a cooldown to not turn multiple times unlike smooth turn. Similar to teleporting, instead of rotating in place, snap turn can fade to black and make it easier for the user to transition between rotations.

The physical movement option is automatically integrated based on how VR and Unreal Engine work.

The user will have the option to rotate with snap or smooth turn, or just turn their head, and move with the option to simply walk to a location, teleport to it or use the smooth movement. This all provides users with many options or combinations to ease into VR and customize.

### 4.3 Design of Whiteboard

Figure 4-4. The text/image drawn in yellow and red combo was drawn with the marker in-game.

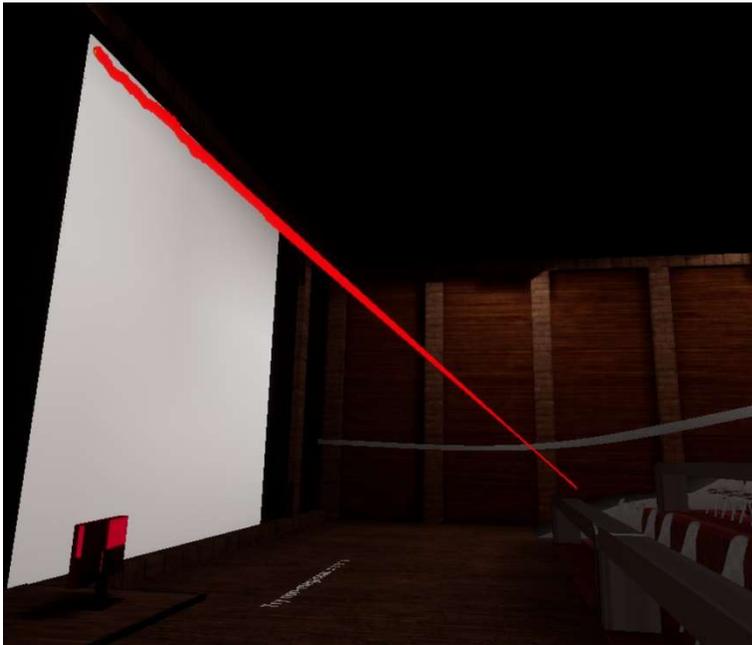


To design the whiteboard, it is a twofold approach. Designing a whiteboard class and a special material to use with it. The mesh for the whiteboard can be a simple rectangle but the material is more complicated. A simple approach is essentially setting up a 'mask'. An additive marker material is created that will overlay on top of the original whiteboard material. To do this, a render target is used to store the information of where to draw the marker on the whiteboard and enter that information into a 'mask' of the whiteboard material. This approach creates an editable mask in function that can be removed and updated without losing the original background of the whiteboard material, by only changing the render target data. The render target informs the whiteboard material where to draw self, and where to draw a mask

of the marker material to overlay it. This means it will be easy to annotate pdfs as well. Further development may be done to accommodate different marker sizes and colors.

A linear algebra algorithm is needed to understand where to draw on the render target. A line trace is appropriate as a common approach in game development for pointing problems. The line trace will take a start and end point and return the first collision. The starting point will be simply the user's hand, right or left whichever they use. But the ending point must be in the direction of where the hand is pointing towards and a set distance away. To do this, the forward vector of the hand is retrieved to get the direction, then multiplied by a constant which we can call  $x$  for now, to increase the magnitude. Adding this new vector to the world location of the right hand gives a point  $x$  units away in the direction of where the right hand is pointing. A line is drawn between these two points, if the line hits any object between the points, it causes a collision check, and the line trace stops there. On the collision check it will check if it collides with the whiteboard and if so, retrieve coordinates of collision on the whiteboard, then tell the whiteboard class to draw on the render target at the coordinate of collision. The value of  $x$  is chosen by the distance from the front desk to the top of the whiteboard.

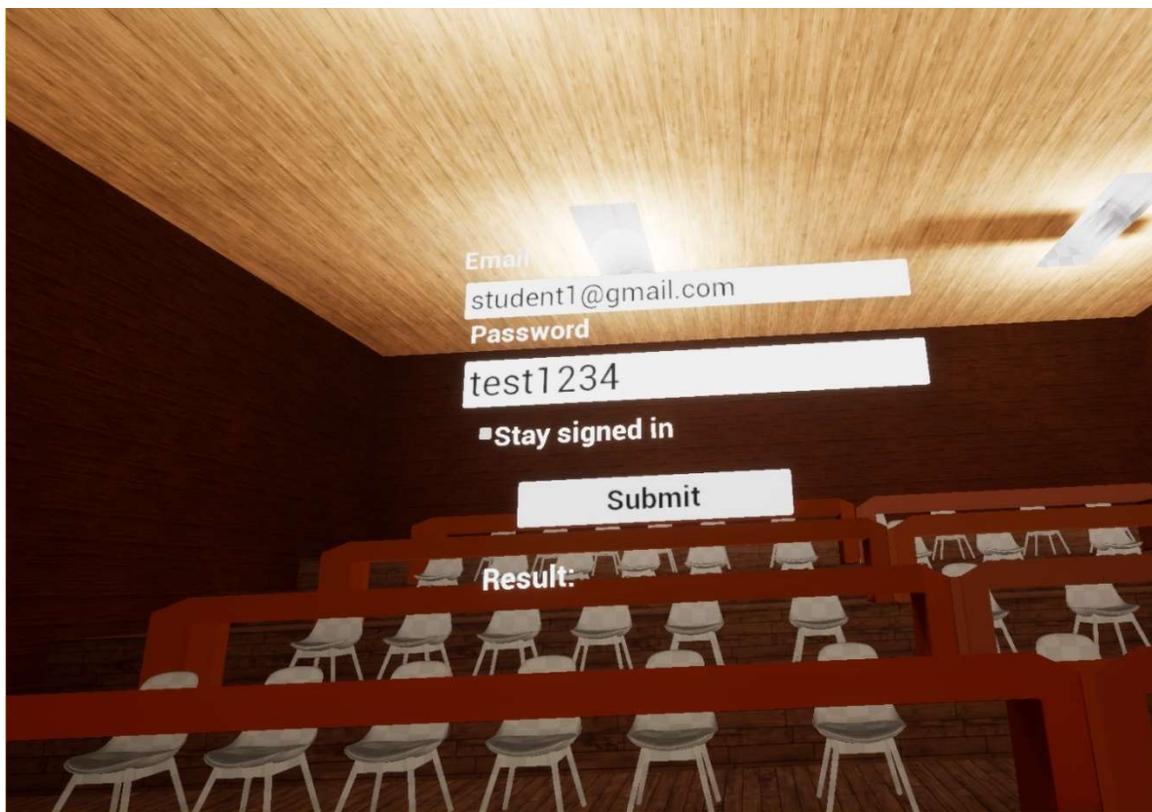
Figure 4-5. Line trace of 1000 units starting from tables to top of whiteboard.



In figure 4.5, red lines are used to show the line generated by the trace. The starting point was the user standing with the back to the table and pointing at top of the whiteboard. The lines connected with whiteboard successfully as shown. 1000 units represents the maximum distance from the ground floor to the top of the whiteboard, hence  $x$  is 1000. The whiteboard material is then enhanced by having the background of the whiteboard be the pdf page chosen by the teacher in class.

#### 4.4 Design of UI and Networking Setup

Figure 4-6. Login screen that communicates with the LMS database



On startup, the VR application will ask for the user's credentials. These will be passed to the API for the Learning Management System to retrieve the user's details. These details will be then used to show the name, registered classes and more. When the user wants to join a class, they can see what classes they are registered to and what classes are currently happening. (This list of class and joining will be done by the servers outside this FYP)

Figure 4-7. Class selection interface

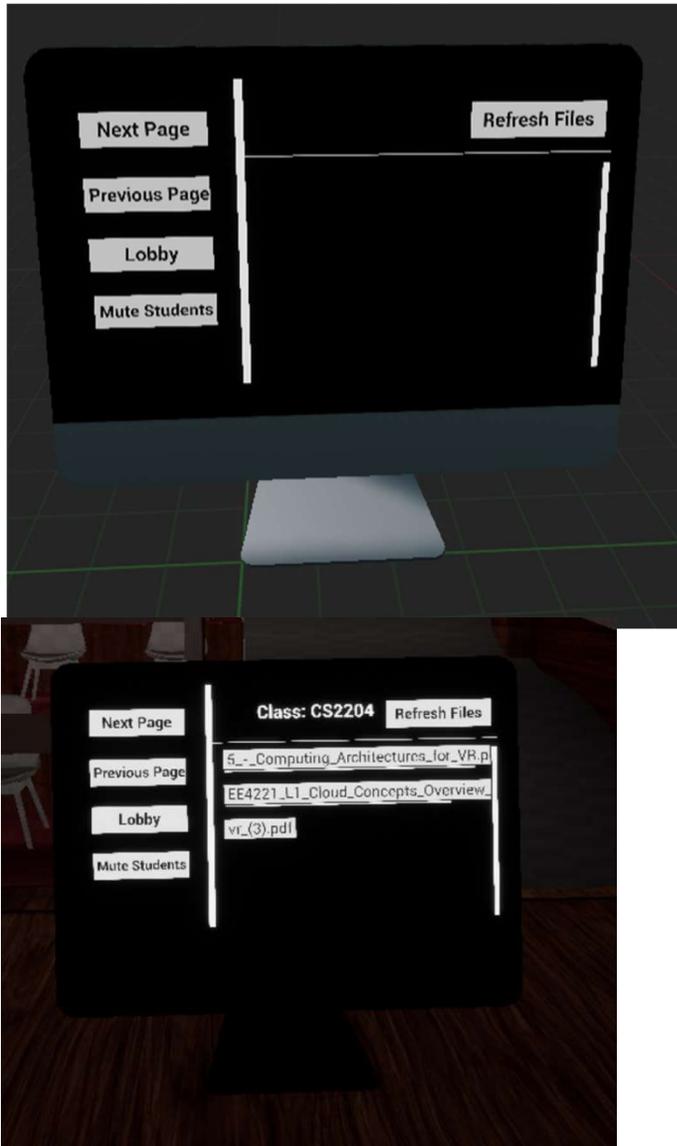


Once they select which class, the information for the class will be passed to the networking servers to connect the user which is out of the scope of this FYP. It will load into the VR classroom that is hosted online. Once the user is connected to the VR Classroom, they have options to adjust movement, audio, and other settings. For this FYP, the UI lets the user pick between registered classes and 'enter' them locally without multiplayer.

#### 4.4.1 Virtual computer Interface

Teachers will be able to use a virtual 'computer' inside the classroom to alter the classroom settings and download files. This interface will be simple containing minimal menus and direct access to options based on Hick's (Soegaard ,2020) and Miller's (1956) Laws of design. The focus is on not displaying too much information at once and keeping a familiar pattern as accessibility and usability. This computer interface will contain any expansion to the classroom and must be designed to accommodate future options.

Figure 4-8(a) and Figure 4-8(b). Virtual computer in editor view (Figure a) and alpha version of virtual computer in use in a class (Figure b). Screenshots from headset are often blurry.



The interface in Figure 4-8 keeps simplicity by having all the class settings on the left side and the downloadable files in the right. This interface contains the most necessary options and does not need sub-menus. In future expansions, the UI can accommodate more options if needed and the computer uses a familiar UI design to the previous UIs. Mute and Unmute students can be used from the same button by having it be a simple flip. Note that the functionality of these buttons is out of the scope of this FYP.

#### 4.5 Design of audio setup

Voice chat is an important feature of the VR Application. There are two types of chatting setup used for the application, 3D audio and direct 2D audio. 3D audio means it will follow the regular laws of sound originating at a single point in 3D space and dropping off farther away the listener is. For '2D audio', the audio will be heard at full volume regardless of location. This way the teacher would not need to be loud to make sure all students can hear, and act as a microphone to enhance their volume. The teacher will be given the option to use 3D audio and 2D audio. If the teacher is talking to a student individually 3D Audio may be preferred, or 2D audio while addressing the entire class. Similarly, the teacher will have the option to enable 2D or 3D audio for students as well or disable their microphone input entirely if it is distracting. This will allow the teacher to customize the class suited for their current situation with all these options. The audio itself will be recorded with the onboard microphone on the MetaQuest 2 and played back from the headset speakers. However, the implementation of this falls outside this FYP and will be handled by another member. For sound effects of different effects such as click effects, are played as 2D audio, and drawing sound effect is 3D audio originating from the drawer's location.

#### 4.6 Quality of Life features

There are small quality of life features throughout the classroom to make it feel more user friendly and interactive. A key of good user design is feedback on actions, and this is implemented here. While drawing a small sound and haptic feedback plays on the controller using a sound wave to design the haptic. While hovering over text box on login, it turns blue to acknowledge the hovering. When changing the movement setting, a small file with the movement mode is stored and used whenever the setting is changed, or the player is loaded. When hovering over UI menus or interactable objects, a visual line will be shown from the user's hand to show what they are pointing to, or where they are drawing. When clicking on buttons a small sound will play to acknowledge the action so the user is not wondering if their action worked. While loading the file list, the computer shows a small loading visual to show the files actually refreshing. The hands play a small animation when the grip button is clicked and have them close depending on how hard the button is clicked. These small features will

help the user immerse themselves and feel like they are actually interacting with the classroom. Without them the classroom feels more dead and artificial to the users.

#### 4.7 Implementation of Learning Management System

This section will cover the implementation of the website, backend service and the API. While implementing the backend service and API, certain issues arose. The Admin API is always signed in as an admin and cannot sign-in users, so the website will have to directly communicate with firebase to sign in users. After testing, the VR Application could not implement the firebase SDK due to environment issues, so a new API is required for the VR Application to indirectly communicate to firebase. The original API will be referred to as the Admin API and the new API will be referred to as the VR API. The VR API needs to access the native SDK for firebase, so node framework is an appropriate choice for this API. It can manage different types of requests, processing, receiving, and sending and be scalable and is supported as a firebase client. All APIs are designed to handle requests concurrently from different users, so REST design pattern is ideal for API design. Requests accessing the same resource will always be structured the same way every time, and every request is processed without any additional session data, only using the request content. Each API is expected to concurrently service different platforms and users, hence managing sessions would add complexity in designing and time in execution. Designing each request to be stateless was instead the goal.

Figure 4-9. High level design of the Learning Management system and integration

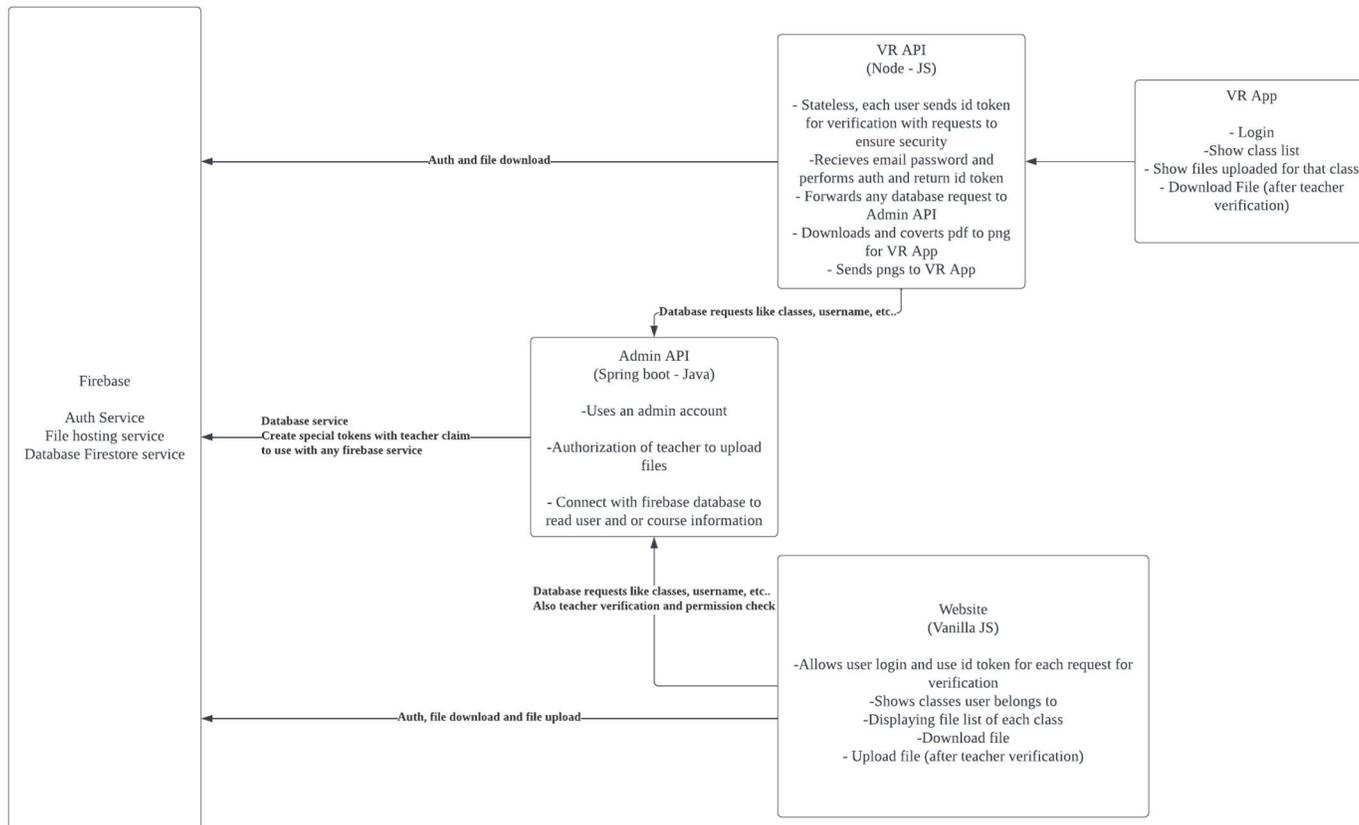


Figure 4-10. Use case diagram of VR API

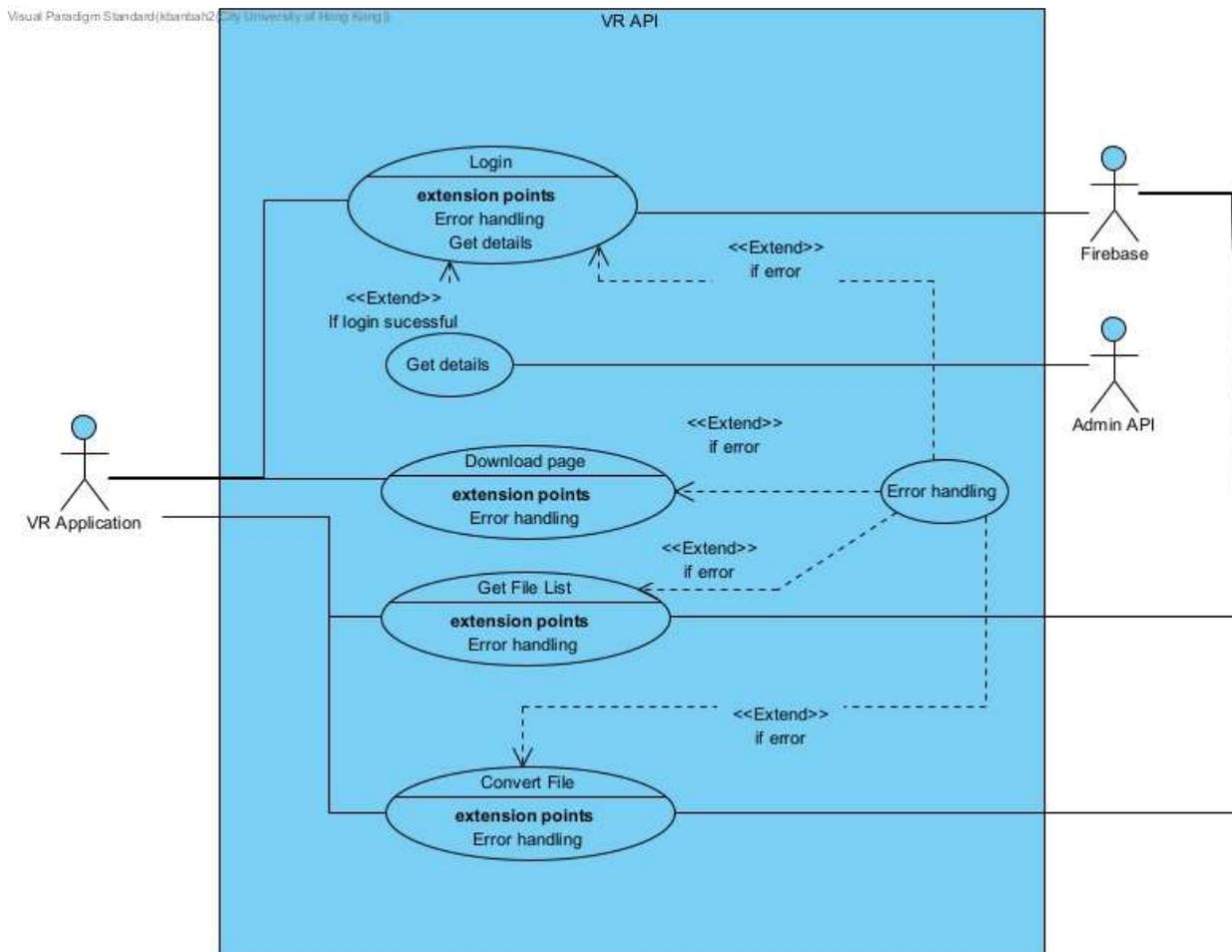


Figure 4-11. Use case diagram of Admin API

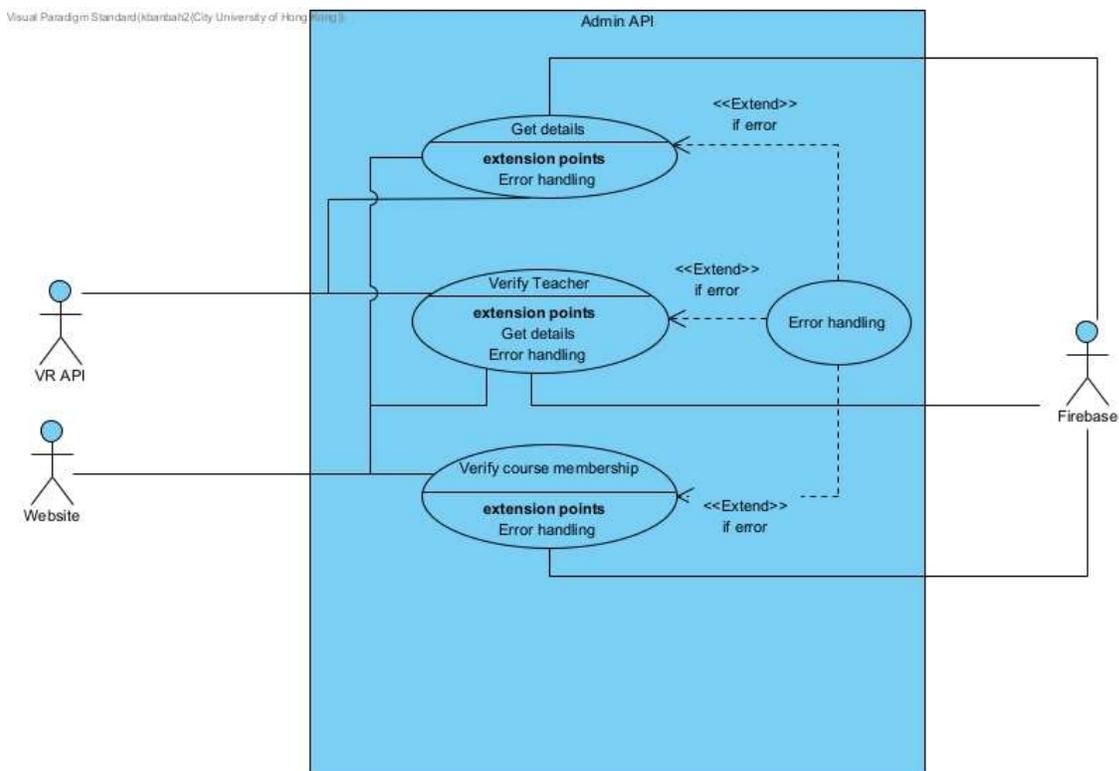


Figure 4-12. Use case diagram of Website

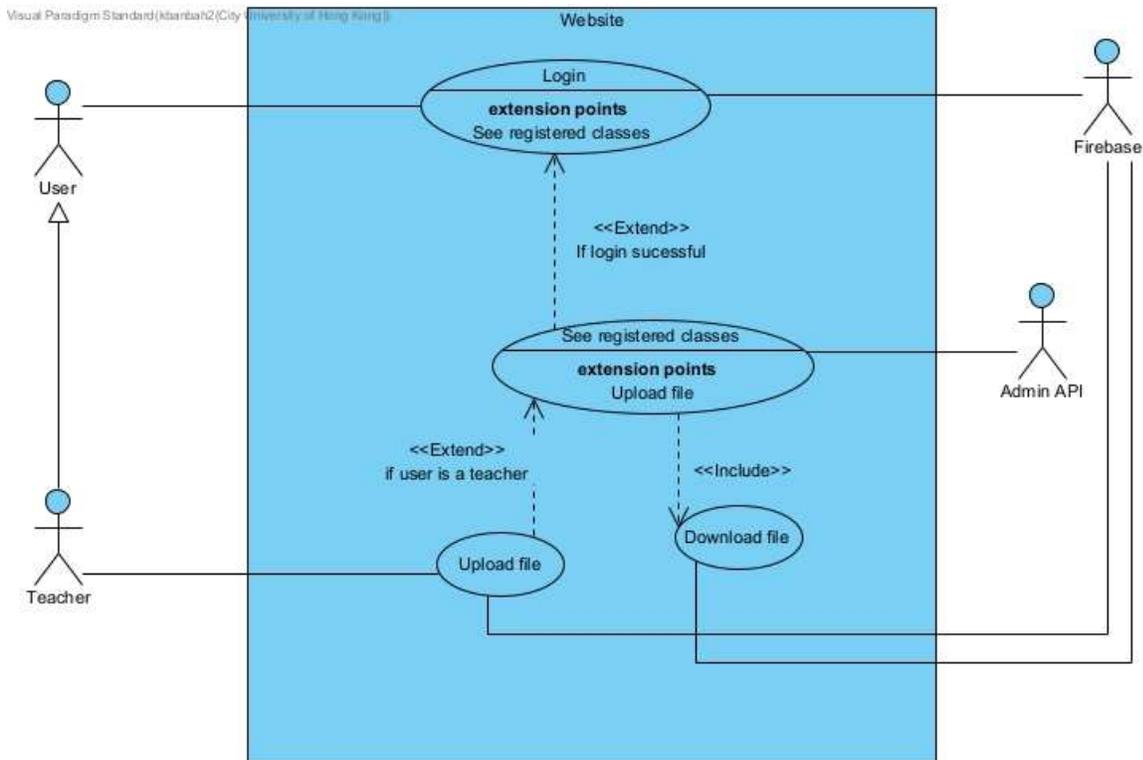
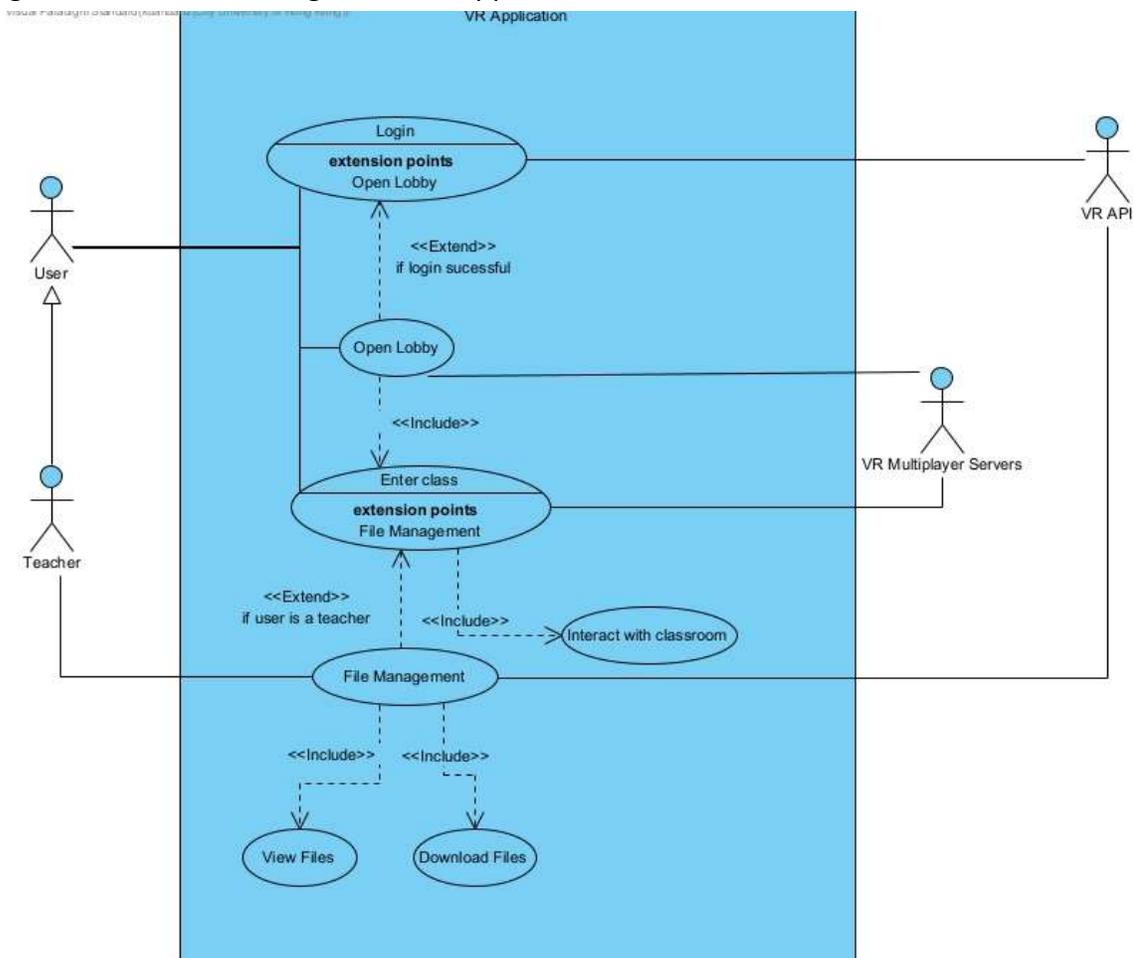


Figure 4-13. Use case diagram of VR Application



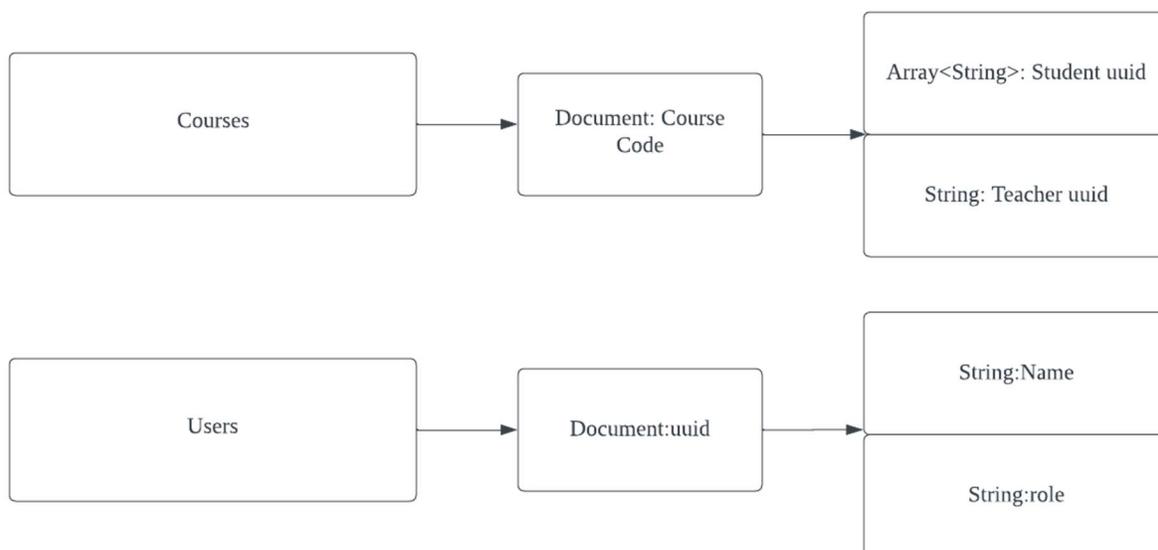
#### 4.7.1 Backend Services – Firebase

Firebase is responsible for facilitating database, authentication, file storage and authorization for different roles. Each of these components are setup separately under the firebase platform and have packages available in SDK for different languages/frameworks.

##### 4.7.1.1 Database

The ‘firestore database’ service contains the user and course data. The firestore database is document based with NoSQL design style and is suitable for the data. The firestore database contains documents which contain fields which contain primitive datatypes like string, numbers, etc.. or other nested documents. The database structure designed with two high-level documents; users and courses.

Figure 4-14. Layout of root level documents.



The users document contains a document for each user titled their uuid (Unique User Identification, a firebase generated id for every user created on sign-up). Each of these sub documents contain the user’s role (Teacher or Student) and their name. The Courses document has course codes for document titles and each document contains an array of student uids and a teacher uuid field containing the teacher uuid. There were other options possible for database design. The user document could contain the classes they belong to instead of the

courses document, but then to find all students in a class it would require searching through every single user and going through their list of courses. A LMS would have more users than classes, so storing all this information in this way was preferable so it would be less time consuming to search all the classes a user belongs to. However, this is still an expensive query, another alternative would be to implement user document containing courses they belong to and courses containing students, and have repeated data to reduce query time, but it would take more storage and introduce management of linked data. If the records were not updated together, it would cause errors. This is a tradeoff of storage and possible errors against query time and higher query time was chosen since it performs reasonably well in tests and is less error prone.

For security, instead of setting up roles for who is allowed to read what information about which source, the admin API is the only way to access the firestore database. No client application can read the database but can instead only request the admin API who can return the user information without any security issues.

#### 4.7.1.2 Authentication and Authorization

Users have email id and password combinations to access their accounts. Different register and login methods can be implemented but for a learning management system it is sufficient for now. On login at the website, firebase will send a response that contains an id token (a string that identifies a user to firebase). This id token is valid for one hour and can be refreshed with a refresh token that is valid for longer. None of the APIs will ever see the email password since only id tokens are passed after login and are used to access firebase services. The VR API will receive email password from the VR Application to pass it to firebase however but that is unavoidable. These ID tokens allow for security since they are valid for a short time and prevent password and email id from being passed more than needed. Even if an ID token was intercepted, it would not be valid for too long. The concept of tokens can be expanded to have custom 'claims' attached as well, for example adding a teacher claim to let the service know the user is a teacher. Once a user's role is verified, this claim can be added so the rest of the services will not 're-query' to verify if the user is a teacher or not.

#### 4.7.1.3 File Storage

For file storage a bucket (root-level folder) is established with subfolders for each course. Since files only belong to course, this is the easiest method of organization. Specific security rules are implemented so that only authenticated users (any valid account) can view the stored files, and only users with a custom “Teacher” claim in their authentication token can upload files. There is also another rule of limiting file size to 50 MB and pdfs only to make sure nothing bad or extremely large is uploaded. These security rules ensure that file storage is not abused by users or any ill actors.

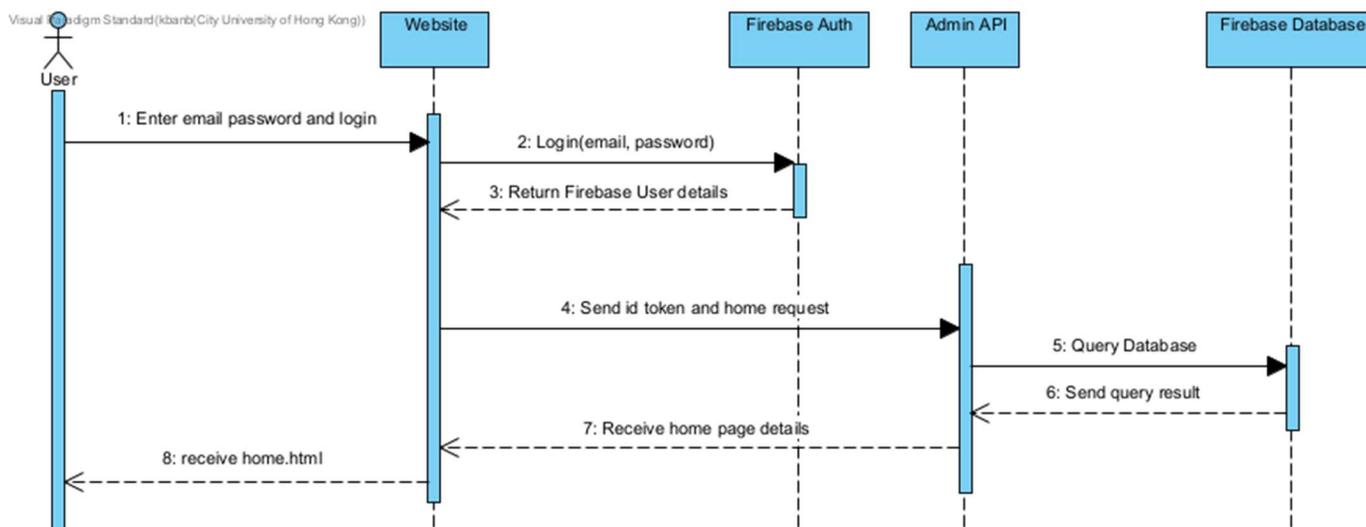
#### 4.7.2 Admin API – Java Spring

The admin API is built in Java Spring to handle requests querying the database. The API has a custom-made class designed for the format of each incoming request. Each request must contain an id token to verify user and the type of request it is making. The id token is for security reasons and request type is to ensure the correct request is handled by the appropriate handler. The first step of every request is to verify with firebase that the id token received is valid and retrieve the uuid of the user who it belongs to. Once verified it can then process the request. To handle all the requests, a singleton database manager class is made. On construction this class reads the admin credentials off a local file that is not accessible to any object or any person because these admin credentials (generated firebase) allow free access to any service without any restriction and hence must be secured. As the API is uploaded as a docker image on a service, it keeps it secure. The singleton pattern is used here because there is no need to re-verify with the database costing additional authentication time each time a request is made. Once the database class is initialized and authenticated, it can just lay waiting for handling any requests quickly.

Another reason Java was chosen was because the way it handles a request is by creating a thread (Servlet Java class) to handle each request. Each request runs it’s own separate thread accessing the database manager concurrently and since each request is separate from each other, the database manager class is thread safe and hence it can run in parallel and expand to accommodate all the request made without needing more instances and instead just

increasing hardware resources to scale up. This design and framework ensure stability, performance and security for the API.

Figure 4-15. High-level Sequence diagram of a successful login loading into homepage



The API handles three types of requests, a home request to get all the basic details of a user such as name and courses they belong to, a teacher verification to verify a user's role by returning a token with the added auth claim if they are a teacher to allow uploading files and the final request is a course details request to verify membership to a course. On an improper request such as invalid credentials or wrong request type the verification fails and returns an error status for the request-maker to handle. For the home request, once verified, the uid is used to query the Course and user document to get the role, name, and courses the user belongs to and to combine them into a single JSON response and return it. For the token request, it verifies the request and checks if the user is a teacher; if they are, an additional claim is generated, and through admin functions in firebase, a special custom token is made. If they are not a teacher, a custom token is still made just without the claim. This special claim will be read by the authorization check by firebase to allow access for uploading. For the course details request, the API verifies, then queries the database to see if the user belongs to that course as a student or teacher and returns the result.

This API is hosted by using docker images on a hosting service called fly.io which offers scalable computing resources should it be needed.

#### 4.7.3 VR API – JavaScript Node

The VR Application API is written in JavaScript with the Node framework. Java could not be used for client applications, only admin SDK. Node.js offers many packages such as express which can manage concurrent requests made at different urls. Making requests in the VR Application require creating tricky HTTP requests in C++ and there is no way for authentication or to retrieve files from firebase at all directly from firebase. To work around this, this API will be the go between and help reduce the number of requests and difficulty for the VR application and manage the parts the VR application cannot do. A benefit of choosing node and express package for the framework is the asynchronous management style. When a request is made, the most time-consuming sections for this application will be when it requests the admin API or Firebase services. With async execution, instead of waiting for the response to arrive, node will instead go manage another request rather than wait synchronously for the response to come. This makes the entire application incredibly efficient since there is very little processing to do with the request responses from other services and it can focus on sending and receiving requests quickly instead of waiting. This style also makes it incredibly simple to scale up into more servers and CPU resources if needed in the future since it is single threaded and stateless when managing any request.

It contains four stateless requests, auth, file download, file list and file convert. When the VR Application sends user email and password, instead of sending a confirmation response with id token and then receiving another request for details, if the login credentials are valid, the API sends a request to the admin API for details and combines that response with id token to send back to the VR Application to reduce the number of requests made by the application.

For the file list it receives the class the user wants to view and id token for verification. Id tokens cannot be used for signing in, and email password should not be sent repeatedly, so a custom token is used to sign in for verification to access the firebase file storage via the admin API. This method keeps security while slightly delaying the request time and is a reasonable trade-off. File download and File convert requests will be explained more in the Showing files in VR section.

#### 4.7.4 Learning Management Website

JavaScript was already being used in the VR API; in the interest of keeping things similar, the website is built in Vanilla JavaScript, HTML, CSS, and Bootstrap as an addition to improve UI elements. One of the first challenges with the website was how to manage user information. Once a user signs in and the id token and details are received, the challenge was how to appropriately store the ID token to have it be accessible to different pages and how to load import the firebase SDKs, make requests and manage performance while loading all resources.

To manage storing ID tokens, there were options such as using local storage in browser or cookies. Cookies was chosen because they cannot be accessed by other sites and are secure. Use case wise cookies are meant for temporary storage with storing only four KB which more than sufficient for id tokens. Combined with the fact that they can be accessed with multiple pages of the same website it makes sense to use them.

The more difficult challenge was how to load the resources for pages such as home and class pages. These pages need to load the firebase services, then their own local scripts to access the services, make requests and so on. The pages also shared a significant amount of code for initial setup, such as the sidebars setup, links, requests, and more. It was easier to create a common setup script that could be imported asynchronously with the firebase services so everything can load in parallel in the browser. Then, the page script can access these services and the setup file to load all the page elements. The difficulty was how to manage async loading while making sure the correct services were loaded at certain points in execution. The script could initialize the functions for creating table rows, for example, but needed to wait for the request to resolve to be able to fill in the table rows. A load resources function is created for each page to load the setup file, make requests, and take in a setup function unique to that page's requirements. All of this was done to keep performance and manage stability and required testing to ensure it worked.

The website on the backend works similarly to the VR API, where it connected directly to firebase for file and authorization and left all the database queries to the admin API requests. For the file management, it needed to verify if the user was a teacher or not to create the

upload button and elements, and handle possible improper permissions, improper file type and or file size. It used the custom token from the Admin API to know if the buttons for uploading files should be generated. If a user creates the button manually by editing the HTML, without the valid token the upload function would fail the security check on the firebase side. The other challenge was to manage downloading and uploading. JavaScript and HTML offer decent solutions for allowing users to select files and allow limitations on file. For uploading files using the inbuilt file type option and the uploading of bytes function directly to firebase. A user feature was added by adding an uploading message that notifies the user of the progress via callbacks. This gives feedback to the user, letting them know the website is working and not just frozen. For downloading the file, a blob was downloaded directly from firebase. The blob is first saved into the browser memory before being written to a file locally. Using blobs means metadata like filename and size are included automatically unlike bytes, but there is no straightforward way to write a blob into the normal file system, so a clickable link with the URL is generated from the blob, that is then clicked internally and it downloads the blob. This is a roundabout method and difficult to come up with, however it is negligible in performance or memory compared to using bytes since the clickable link is immediately destroyed and takes little memory and sets the metadata and file name automatically.

#### 4.7.5 Security

Overall, there are some sensitive details being passed through with requests and being stored locally that need to be properly secured. The first line of security is the HTTPS protocol usage. All APIs are hosted on HTTPS servers ensuring encryption on any requests received or sent, firebase and unreal engine both use HTTPS for communication as well. Security rules are used for each firebase service to not let unauthorized and unauthenticated users access to resources. Additionally, POST requests are used for every resource including an id token so data can be sent in request body adding a little more security. And as mentioned previously, the entire system uses tokens that are only valid for an hour and will then need to be refreshed which is allowed for 24 hours without a re-login process. Additionally, creating special custom tokens with claims to check if a user is a teacher or not and allowing adding of any other special

claims is more secure than sending true to check if a user is a teacher but rather adding it in an irreversible way to the user's token that cannot be worked out easily.

## 5 Showing files in VR

A major technical difficulty in the project is to create a method to download user uploaded pdfs and show them in VR. This problem has multiple challenges such as downloading the pdf onto the headset and converting it into a format that can be read by unreal engine and loaded at runtime as a texture. Firebase does not offer a C++ SDK that can be used in Unreal Engine, so the files must be passed through an interim in the VR API.

### 5.1 Creating textures at runtime

The first challenge was to load files as textures at runtime so they can be added to dynamic materials in unreal engine. This is only possible for certain file formats like pngs, jpgs and other image formats. To accomplish this, a few things are needed. The file needs to be read into an array of bytes which are read by an image wrapper. Then using pointers and copying directly from memory, a new texture can be created that reads bit by bit to create a texture that can be loaded directly. Because of reading directly from RAM, there needs to be some safeguards and a mutex lock is used for safety to prevent any memory errors. This is a low-level solution that is quick and successful in converting images to textures. This texture can then be loaded into a dynamic material at runtime as a texture parameter. This dynamic material is the whiteboard material mentioned earlier, but the background color is now a texture parameter. An instance of the material is generated at initialization that is converted to be dynamic and updated whenever the user requests a file.

### 5.2 Sending the file to the VR headset

The application can only take image formats, so the files need to be converted page by page into image to send to the VR application. To achieve this, the pdf must be downloaded onto the API first, then converted and sent to the headset. To achieve this the VR API first converts and stores all the converted files in a folder and whenever the application requests a new page, the API will respond with the converted page. The issue with this approach is that it

takes time to convert a single file into multiple pages and the time it takes to download the pdf. However, this saves space because pre-converting all files will take too much space to store on the API. This is a compromise on resources on space vs duration of request. The file convert request will entail the VR API downloading and converting the file locally and the file download request will respond with the converted png that is then converted to a texture.

## 6 Testing Plan

To test an application like this, the testing will be split into two types, performance, and quality. Performance will focus on the raw performance of the system, while quality will focus on ensuring functionality and correctness of the applications.

### 6.1 Performance Testing of VR Application

Performance of the application is essential to a good experience. This begins with the 3D models and materials that are used to design the application, to shadows, lighting calculation or prebuilt and other features. These are all in app settings that will affect the performance.

Exporting and selecting the correct rendering setting is important. To export the entire application to the headset it requires optimizing a process that begins with android SDK tools using android studio version from specifically 2020. Then it moves to using specific .NET libraries and packages from Visual Studio. Then, Unreal Engine needs specific scripts to run for android setup. Once this done, the Oculus app can be used to monitor real-time statistics and performance. This entire process required intense research and testing to figure out which SDK version and tools were needed and the specific .NET libraries and packages from Visual Studio for game development.

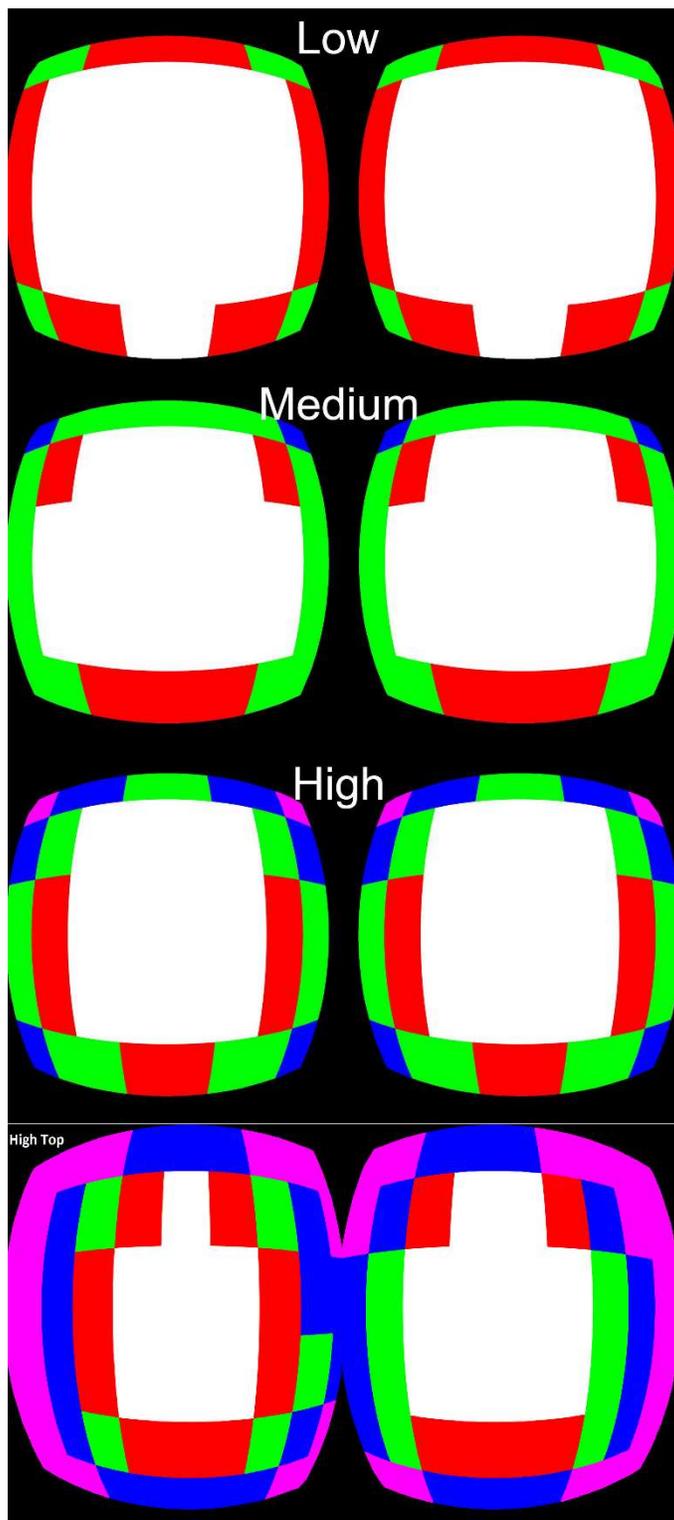
There are two areas that can be focused on for performance optimization: design and rendering optimization. Design includes anything design related such as model, material, lighting, code runtime and similar aspects. For lighting, dynamic, static and shadows all significantly affect performance. In Unreal engine, shadows in VR require a special rendering technique called Cascaded Shadows (CSM), but it only creates them for directional lights, i.e lights like the sun, instead of point lights which the classroom uses. Points lights are a light

source which originates from a single 3D point and have attenuation spheres and fall off pre-configured and customizable. They also can have their shapes changed to have the source be a cylinder similar to a tube light and simulate the lighting of that. Since shadows are not being used due to platform limitations, all the lights are static saving significant performance. The material and models used, have their quality managed globally by the engine and are not recommended to be tweaked further unless necessary to not lose visual fidelity.

On the rendering side, refresh rate, FPS and Foveation Levels are important factors. Refresh rate is the rate of screen updating per second, ideally, this is the exact same as the Frames Per Second (FPS), if not, screen tearing or stuttering occurs. There are multiple refresh rates supported by Quest 2, 72,80,90 and 120. The application can switch between these and make them the performance goal. FPS is a count of how many frames per second is the hardware able to generate. This number may differ from refresh rate at times, but as long as it is not consistently or for long durations different than the refresh rate, then it can be acceptable.

A rendering technique that is used is Foveated Rendering. Foveation Levels work by reducing the resolution of a certain parts of the screen and hence saving on performance (Meta Quest, n.d). Higher levels reduce the resolution more and result in more performance. In bright and high contrast environments, it becomes more obvious if users look at their sides of the screen more often and is not desired. There are three types of foveated rendering that are implemented by meta and unreal engine: dynamic fixed foveation, fixed foveation and eye tracking foveation. Eye tracking foveation tracks the user's eye to dynamically adjust low- and high-resolution zones based on where the user is looking, this is only supported in Meta's Quest Pro but not Quest 2 so it cannot be used here. Fixed Foveation is setting a level of foveation from 0-4 levels and having it fixed for the entire application. Alternatively, dynamically fixed foveated rendering, which adjusts the fixed foveation level based on the current FPS and refresh rate. If the FPS is too low, then increase Foveation levels to compensate and so on. This can be limited by setting a max foveation level as well.

Figure 6-1. Fixed Foveation levels by Oculus



(Meta Quest, n.d)

In Figure 6.1, the different foveation levels in Meta Quest 2 are visualized. White represents native resolution, red is 1/2, native resolution, green is 1/4, blue is 1/8 and magenta is 1/16. In eye tracked foveation levels, these zones are not static and move according to where the user is looking. The different levels low, medium, high and high-top correspond to the 0-4 levels where 0 is no foveation and 4 is high-top setting.

The goal of the stress testing of the headset is to establish the maximum users it can handle rendering at what refresh rate. The minimum goal is to support at least 30 users at 72Hz (refresh rate). Each test is a 5-minute test where the player enters a classroom, downloads a file, draws and walks around the classroom. During this, the simulated users, if any, are rendered and going around the room. The player will also then change pages and download other files to stress the CPU as well. The primary focus is FPS, in addition to FPS, the CPU, GPU utilization and level are tracked to determine any problems. These are tracked using OVR tool by Oculus to profile app performance.

First a baseline test with no simulated users will be done at 120 Hz to test the highest possible refresh rate and max performance with no foveation to see how well it performs just by itself without extra users.

#### Testing conditions

- Users: 0
- Refresh rate: 120Hz
- Foveation setting: Disabled

Figure 6-2. Graph of refresh rate and FPS over 5 mins. Red line is the refresh rate, pink is FPS.



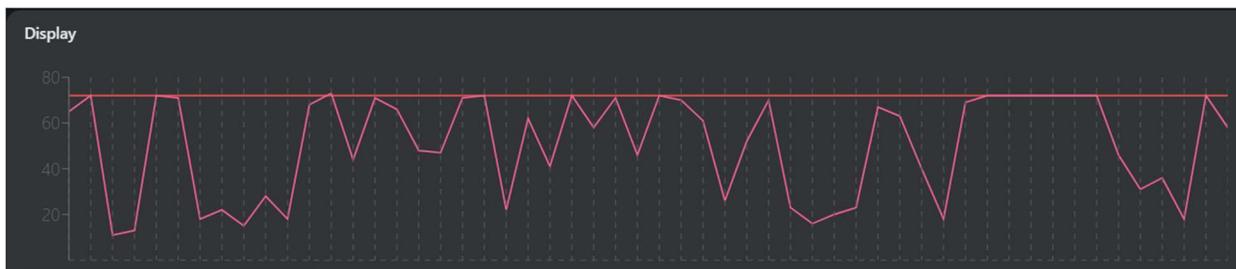
In Figure 6.2, the performance is consistently high and at 120Hz. The dips are when interacting with the UI and loading text occasionally. Overall, the base application seems to perform well without any optimization whatsoever and is relatively successful in the test.

Now the original ideal conditions can be tested.

### Testing conditions

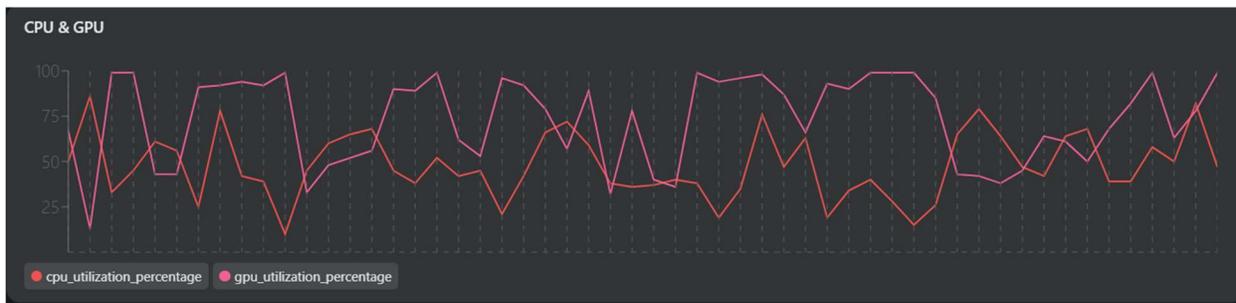
- Users: 30
- Refresh rate: 72Hz
- Foveation setting: Dynamic with maximum level 2

Figure 6-3. Graph of refresh rate and FPS over 5 mins. Red line is the refresh rate, pink is FPS.



An interesting thing to note is how GPU utilization drops at certain points. This is due to occlusion culling, which is a rendering optimization to not render objects that are not visible instead of rendering everything all the time. While the player is not looking at the simulated users, the GPU utilization drops though the FPS does not decrease.

Figure 6-4. CPU and GPU utilization, pink line is GPU usage and red line is CPU usage.

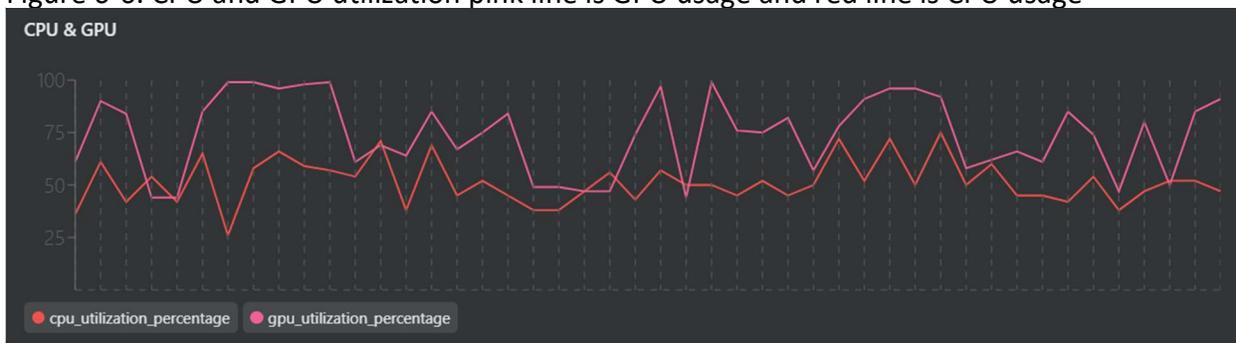


This test clearly shows a significant amount of frame drops during certain sections, while testing, these occurred when the simulated actors were moving. The foveation level switched from 0-2 during more intense sections to bring some performance back, albeit not enough. The GPU load is still too high from the utilization graph in figure 6.4. For the next test, ambient occlusion, shadow related passes are all disabled to lower amount of GPU load. The lightmap resolution of models like the ceiling has been reduced as well. These will remove unnecessary passes since shadows are disabled and lower lightmap complexity will lower the demands on GPU. Lightmaps are a sort of texture that defines how the light should affect the model. Higher detail lightmaps with more resolution are more performance heavy but look nicer, lower detail are the opposite. If the next test is unsatisfactory, the number of simulated users will have to decrease.

Figure 6-5. Graph of refresh rate and FPS over 5 mins. Red line is the refresh rate, pink is FPS.



Figure 6-6. CPU and GPU utilization pink line is GPU usage and red line is CPU usage



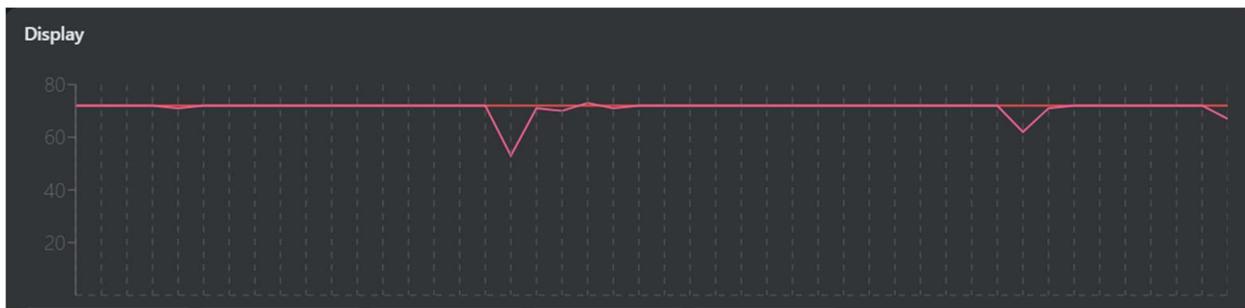
From figure 6.5, the performance is significantly better and the GPU load in figure 6.6 has fallen significantly. At these testing conditions the performance is nearly good enough. In the next test, the maximum foveation level will be increased by one to compensate for the performance. In games like *Beatsaber*, foveation levels go up to 4 from testing, setting a

maximum of 3 with dynamic foveation should be acceptable and result in a successful stress test.

#### Testing conditions

- Users: 30
- Refresh rate: 72Hz
- Foveation setting: Dynamic with maximum level 3

Figure 6-7. Graph of refresh rate and FPS over 5 mins. Red line is the refresh rate, pink is FPS.



In figure 6.7, the FPS is far more stable with only occasional dips while loading new levels or other one-time actions. At 30 users, 72Hz is achieved and is stable and tested. Now alternate testing conditions can also be tested to see more dynamic arrangements of classroom.

#### Testing conditions

- Users: 20
- Refresh rate: 90Hz
- Foveation setting: Dynamic with maximum level 3

Figure 6-8. Graph of CPU and GPU utilization and Graph of refresh rate and FPS over 5 min, red line is refresh rate, pink is FPS.



In figure 6.8, both CPU and GPU are consistently not at peak usage and FPS is almost always at 90 Hz. The application can successfully run at higher refresh rates with lower classroom size. With dynamic foveation, the levels shifted from 0-2 regularly with occasionally reaching 3. Not all the simulated users are always visible, hence via occlusion culling the load is not constant and has peaks and lows. An issue this brings out is that the user avatar design is too complex and heavy on the rendering, better optimized avatar design would improve results.

Currently, these results show the application can benefit from a dynamic adjustment system of refresh rate depending on classroom size. However, this system is out of the scope of this FYP and will need further development to create a balancing algorithm to dynamically adjust the rate.

## 6.2 Performance Testing of Learning Management System

The performance testing of the learning management system will be testing each separate API on how much resources it uses for a single instance, scalability, bottlenecks and turn-around time when dealing with requests. The website will be judged on how quickly it loads content and time to interactivity.

To test the APIs, JMeter will be used to simulate multiple users concurrently, set time limits for each request and response checking to ensure request is properly handled. Both APIs are hosted online at Fly.io with specific hardware limitations. They can be scaled by either increasing CPU or RAM, or by increasing the number of instances of the APIs managed by a load balancer to balance number of requests.

For the website, Google lighthouse metrics will be used to judge interaction and available developer tools to analyze performance.

### 6.2.1 Admin API Testing

The Admin API has three types of requests it manages.

- Creating a custom token for teachers,
- verifying if a user is in a certain class or not.
- Getting all the courses a user is in and their name. (Home page for the website)

For testing, the most resource heavy or time-consuming request will be tested. The request with the most resource usage is in the third request and will be used. It takes most resources because it needs to query all courses and users for getting teacher names, course names and user's name and is used anytime a user logs into the website or the VR application.

Testing conditions:

- RAM: 512 MB, CPU: Single shared CPU
- Each request must be handled within 1000ms or be classified as a failure.
- Each request must return the correct response or be classified as a failure.

After stress-testing, at 600 requests concurrently over a single instance, the testing setup cannot create 700 requests concurrently without errors and needs to start spreading requests over multiple seconds to prevent socket and thread issues. At 600 requests, the single instance can sufficiently perform and pass without errors.

Table 6-1. Stress testing of Admin API

| Requests | Error Percentage | Average Response time | 90% response time | Max time |
|----------|------------------|-----------------------|-------------------|----------|
| 600      | 0%               | 74ms                  | 84ms              | 135ms    |

At 600 requests, 90% of the requests are handled within 84ms, with the longest request only being 135ms to service. To handle more requests even faster, multiple instances can be used of the API to balance the requests, but a single API can easily handle 600 requests without any issues. It is possible for it to handle more but cannot be simulated without errors. For real-world use, the API can be scaled up in resources as in CPU and RAM or in instances to adjust for more simultaneous users. The API performs well and can handle reasonable real-world situations.

### 6.2.2 VR API Testing

The VR API services four types of requests

- Login
- File List
- File download
- Download page

Login cannot be stress tested because of authorization limits with the Firebase service. If too many requests originate from the same location, i.e. 100 or 200 in one second it will exceed allowed quota. The other requests will be tested instead.

Testing conditions: Download file

- RAM: 512 MB, CPU: Single shared CPU
- Request a file

Because node is single threaded, only a single file at a time can be downloaded, hence the performance of converting one file will be tested. The tested file will be a 73-page pdf of size 4.56 MB to reflect real use cases.

Table 6-2. Time taken to convert file

| Trial 1 | Trial 2 | Trial 3 | Trial 4 | Trial 5 | Average |
|---------|---------|---------|---------|---------|---------|
| 888ms   | 970ms   | 976ms   | 843ms   | 989ms   | 933ms   |

This is the time it takes for the API to convert the file into 73 pngs to let the VR Application download. With larger files with more pages or pictures this will take longer, but with stronger hardware that will also be reduced. This establishes a baseline on API performance; however, concurrency here will be weak since this is CPU-Intensive, node will optimize the file writing, but the conversion is performed via CPU.

Testing conditions: Download Single Page

- RAM: 512 MB, CPU: Single shared CPU
- Request individual png (pages of a pdf)
- Each request must be handled within 1000ms to not be a failure.

Table 6-3. Stress testing of download page request

| # of Requests | Error % | Average time | 95% line | Median Time |
|---------------|---------|--------------|----------|-------------|
| 600           | 0       | 130ms        | 405ms    | 33ms        |

The strength of managing I/O operations in Node is clear here. Up to 600 classrooms can request a single page to change in their file and download a png in half a second. This is acceptable latency and good enough to manage concurrent classrooms.

Testing conditions: File List Request

- RAM: 512 MB, CPU: Single shared CPU
- Request list of files in a class

Table 6-4. Stress test of file list request

| # of Requests | Average time | 95% line | Median Time | Maximum time |
|---------------|--------------|----------|-------------|--------------|
| 100           | 784ms        | 1182ms   | 697ms       | 1466ms       |

This request is slower than the file download request, it relies on a query answered by firebase which is the cause of the delay. However, with all requests being served under 1.5s, and that it only runs when a classroom is started, or if refresh files is clicked (infrequently), it is not a major issue and cannot be sped up due to firebase limitations. None of the requests returned error other than being slow to execute.

### 6.2.3 Website testing

Chrome devtools will be used to analyze load times of each page. However, there is discrepancy due to different internet speeds of users. Hence average speeds can be tested over multiple trials and will be used to establish an estimated baseline performance.

Table 6-5. Trials for website load speeds

| Page  | Trial 1  | Trial 2 | Trial 3 | Trial 4 | Trial 5 | Average  |
|-------|----------|---------|---------|---------|---------|----------|
| Login | 572.6ms  | 463.9ms | 461.1ms | 472.2ms | 459.4ms | 485.84ms |
| Home  | 1438.2ms | 642.2ms | 595.3ms | 625ms   | 624.3ms | 785ms    |
| Class | 965.1ms  | 904.6ms | 892.5ms | 850ms   | 904.8ms | 903.4ms  |

These tests are not definitive due to the nature of network connections. On slower connections they will take longer to load because requests will be processed slower and have to wait. These numbers are a baseline indicator of the website's performance. Different actions now such as uploading, downloading, logging in will be tested now. A pdf of 4665 KB size and 73 pages was used to test file download and upload speed.

Table 6-6. Trials for actions on website

| Action        | Trial 1 | Trial 2 | Trial 3 | Trial 4 | Trial 5 | Average |
|---------------|---------|---------|---------|---------|---------|---------|
| Login to home | 812ms   | 828ms   | 865ms   | 789ms   | 794ms   | 819ms   |
| Upload file   | 549ms   | 754ms   | 652ms   | 615ms   | 546ms   | 623.2ms |
| Download file | 288ms   | 245ms   | 507ms   | 264ms   | 209ms   | 302.6ms |

These numbers show on average, most actions on the website will take under a minute. Uploading and downloading a large file may take more time, but the website performs satisfactory overall. The ideal goal was to have every action be under a second which has been achieved.

### 6.3 Quality testing of VR Application

Testing a 3D VR application through automation requires tools that do not exist. The complexity of creating an AI to test and attempt to stress and break and expose bugs in 3D application like this, is not worth the effort. The current approach to test games and similar applications is through Q/A testing. Testers receive special builds of test, try to break it, then report any bugs found. This is usually in two stages, alpha and beta testing. Alpha testing is internally testing the application and beta is testing it publicly through either a closed or open beta with volunteers. Through Q/A testing, different bugs in the application were found and fixed. Some of the bugs found are noted here for example. The general methodology was to try all the features in creative ways and try to break them.

Table 6-7. Bugs found in Q/A testing

| Bug found  | Result   |
|--|--|
| Teleporting through the whiteboard   | Setup a blocking volume covering the wall  |
| Teleporting onto the computer table  | Setup a blocking volume covering the area  |
| Teleporting and moving during UI only levels   | Fixed the movement disability and redo-ing the movement  |
| Snap movement rotating too quickly   | Added a minimum cooldown period between snap rotations   |
| Hovering on the whiteboard with one hand created the line from the other hand to show target | Created system of recognizing which hand is hovering and only showing pointing lines appropriately |

Not all bugs in application will be caught but a significant majority at least have been caught and exposed by alpha and beta tests.

#### 6.4 Quality Testing of Learning Management System

All the APIs and the website need to be tested for reliability and accessibility. Each API will have a series of test cases that must be passed to validate correctness.

##### 6.4.1 Admin API Testing

The Admin API is developed in Java using Spring and Gradle. These tools and Java offer methods to create Junit Test cases to verify correctness. Junit cases are made for every request type using Junit library and Springboot Test. The difficulty in testing is simulating a firebase id token which is used by all requests. These tokens are only valid temporarily and would need to be recreated each time a test is done. To overcome this, a new request can be made to an exposed firebase endpoint to create id tokens at runtime for testing. The exact format of this request is specified in the documentation and properly configured HTTP requests are needed. These tokens are then used to simulate users and test the end points of the API. An additional issue with testing was that certain functions in firebase, specifically authentication related, are

not available in the Admin SDK on Java. But only available on the client SDK on platforms such as JavaScript and Node. To test custom token functionality and verify it working, a testing client in JavaScript and Node is developed since that cannot be verified in the Admin SDK and hence the VR API was made. With all these considerations, testcases that generate id tokens are created and an extra testing JavaScript API is made.

Table 6-8. Unit Tests written for Admin API

| Request being tested                  | Test types                                       |
|---------------------------------------|--|
| Home Request                          | Incorrect Request Type                           |
|                                       | Valid teacher user id token                      |
|                                       | Valid student user id token                      |
|                                       | Blank id token                                   |
|                                       | Invalid id token                                 |
| Teacher verification request          | Valid teacher user id token                      |
|                                       | Valid student user id token                      |
|                                       | Blank id token                                   |
|                                       | Incorrect request type                           |
| Verify Membership to a course request | Valid student user id token with correct class   |
|                                       | Valid student user id token with incorrect class |
|                                       | Valid teacher user id token with incorrect Class |
|                                       | Valid teacher user id token with correct class   |
|                                       | Valid user id token with class invalid class     |
|                                       | Incorrect Request type                           |
|                                       | Blank id token                                   |

An important point to note, is due to how requests are handled in Spring and the method of reading request, if the request content or URL is improperly formatted, then an error will

automatically need to be returned. This includes cases such as incorrect request verb or routes or more. The testcases only need to consider blank and invalid fields and similar errors.

#### 6.4.2 VR API Testing

The VR API is the API developed in JavaScript with Node and handles all the requests coming from the VR Application running off the Meta Quest 2. It handles Auth, FileList, File Download and File convert requests. Similar to the Admin API, id tokens are needed for some requests and need to be dynamically generated. The same method as the Admin API does not need to be used for this, since as a client application, directly signing in is possible. For downloading the files, they will be downloaded pngs that can be compared with the same page of the original pdf. To test this, a JavaScript testing framework called chai will be used. It can be added into the npm module list as a testing framework and ran from command line. With it, it can directly connect to the API and can simulate a client.

Each request has complications that must be accounted for. For the authentication request, the course details need to be tested. There is no clear way to compare the entire object, so going field by field is necessary. The generated id token does not need to be included since it is directly created by firebase. For the file list request, id tokens could directly be generated from firebase to simulate users. For file conversion request, there were issues with timeouts as conversion took too long and the test timed out and id tokens again had to be generated dynamically to ensure proper testing. For all requests, the expected object was compared with the actual object; this could be a single message or a JSON object. For the final request of image download, the byte array of the received file could directly be compared to the previously converted file to verify success.

Table 6-9. Unit test for VR API

| Request                 | Test case                             |
|-------------------------|---------------------------------------|
| Auth                    | Test student account                  |
|                         | Test teacher account                  |
|                         | Test false user                       |
|                         | Test invalid user password            |
|                         | Test invalid user email               |
| File list Request       | Test Teacher account                  |
|                         | Test student account                  |
|                         | Test incorrect id token               |
|                         | Test incorrect class                  |
|                         | Test no id token                      |
|                         | Test no fields                        |
| File conversion request | Test valid file path for student user |
|                         | Test valid file path for teacher user |
|                         | Test invalid path                     |
|                         | Test invalid payload – id token       |
|                         | Test invalid payload – file           |
|                         | Test missing payload – id token       |
|                         | Test missing payload – file           |
| Image download request  | Test Valid Image                      |
|                         | Test Invalid image                    |
|                         | Test Invalid route                    |

All these tests ran successfully with appropriate error messages to reflect the problem. Similar to the Admin API, any URL or route or incorrect verb usage is automatically handled by the framework to return an error html page and status code.

### 6.4.3 Website Testing

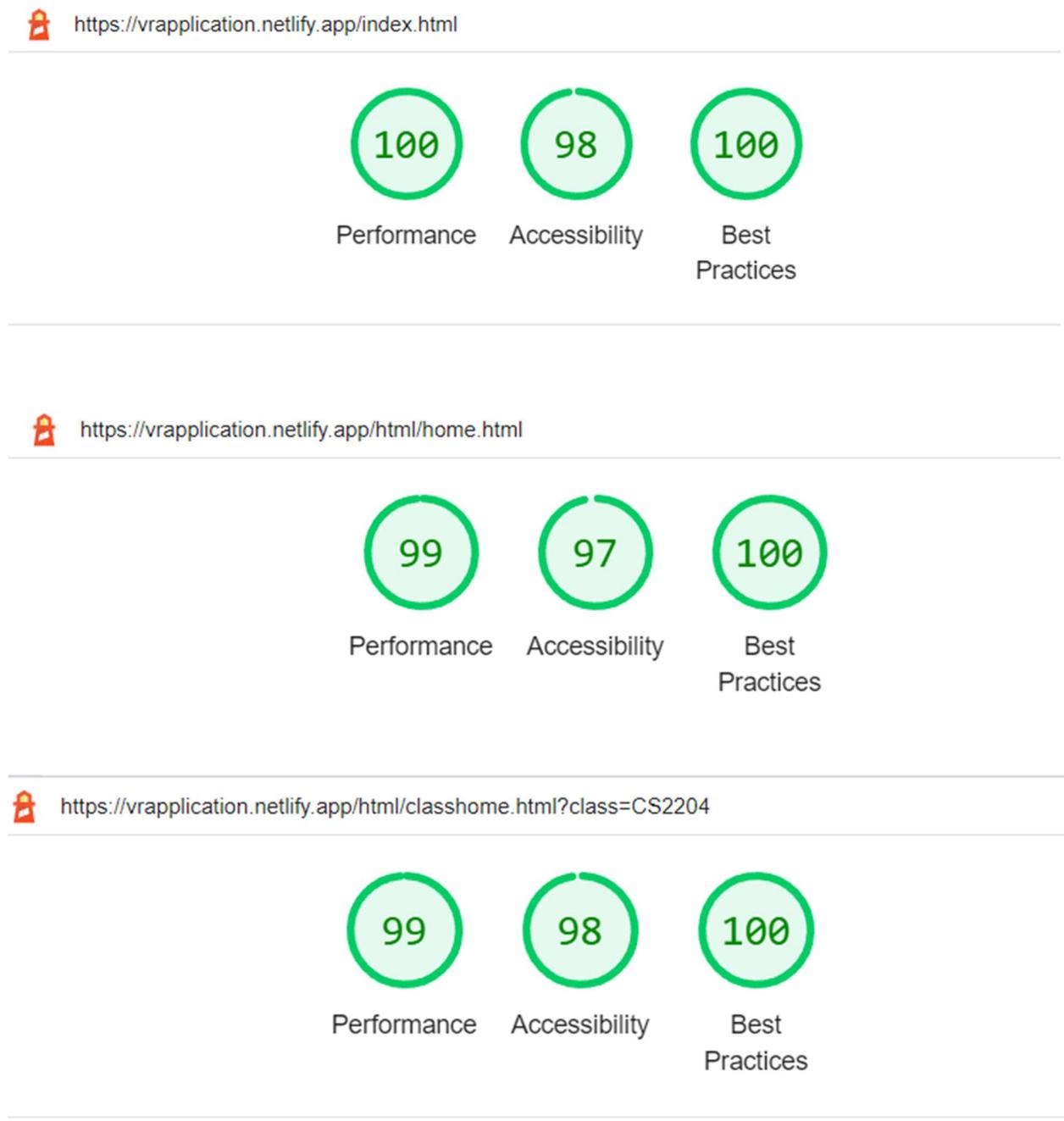
Website quality testing will occur through testing with selenium. Selenium allows simulating user inputs. Testing will focus on trying out edge cases or incorrect inputs.

Table 6-10. Test cases for Website

| Test cases                     | Result   |
|--------------------------------|--|
| Incorrect password/email       | Website alerts to incorrect email/password                             |
| Upload non-pdf                 | Alert saying non-pdf upload  |
| Upload over 50 MB              | Alert saying reduce file size  |
| Accessing non-registered class | Message showing no permission  |
| Log out button clicked         | Returns to login page  |
| Student logs into see class    | Cannot upload or delete files, normal table loads for downloading file |

Google lighthouse, a tool from google to judge performance, accessibility and best practices was used. All pages resulted in passing marks. These tests ensure good coding practices and accessibility features such as labels on forms, heading formats, page titles and more. They also confirm security, checking vulnerable libraries used, HTTPS security, non-deprecated APIs and more. This ensures the website is up to standard in form as well as functionality.

Figure 6-9(a), Figure 6-9(b), Figure 6-9(c). Results of google lighthouse audit on login (index) (a), home(b) and class pages(c) respectively



## 7 Final Results

### 7.1 Results of VR Application

The VR Application is ready to be delivered and attached to the networking subsystem developed by my group mate. The application runs at a stable FPS and is stress-tested. It allows teachers to download files and draw on the whiteboard. The movement and design of the classroom is all finished and sufficient with the original requirements.

### 7.2 Results of the Learning Management System

The learning management system is a complete full stack website with database and APIs that are all hosted. It is secure and can handle errors and restart automatically instead of needing manual input. Performance and quality tests have been performed successfully and passed and the system is designed in a scalable manner. It is functional and the VR Application can sign in users, download files, view courses and more.

## 8 Conclusion

This project is an extremely ambitious goal that was achieved through many technical difficulties and research. VR development is still far from being easy for the mainstream and user acceptance is still not ideal for large scale revolutionary applications but might be soon. There are also many extensions that can be built upon the project to improve it.

### 8.1 Issues

The project had many technical issues and considerations that were not clear at the start of the project. The VR development process itself is not straightforward and not documented clearly. Setting up the pipeline to develop on, which version of software to use, Unreal engine 5, 5.1 or 4.27, which headset to develop on, managing android packaging and specific versions of SDKS and NDKS. Older versions of unreal engine had marginally more support, but they lacked features that the newer ones had, but newer builds were untested and had to be compiled from developmental branch of Meta and Epic for developers. The setup for the project took more time, research and testing than expected. Combining this with the rendering optimizations such as Foveation and lighting management which had to be understood and

applied. Understanding the rendering process of how pipeline of Application -> Geometry -> Rasterization in the context of VR can always be improved and researched deeper.

On the VR Application side, designing the entire movement system to be smooth was a technical challenge, discussed in Section 4. Designing the movement required understanding the vector calculations that go into calculating user location and differentiating between direction and position. For teleporting, projecting a line and testing if it hit a valid location and can be teleported. Fixing all the edge cases of showing the line when it can be teleported, teleporting when it is close to a valid location, and catching every case to prevent the user from being teleported when they shouldn't, and required testing to design a logical algorithm of when, where and if to teleport.

Due to the nature of unreal engine, using C++ to make web requests including setting appropriate headers and json body, callbacks and memory management on a 'mobile' device needed testing to ensure reliability to prevent leaks or overwrites and crashes. C++ is a low-level language and manipulating the memory, reading objects as bytes to process responses or files were all different challenges to go through.

Learning the specifics of VR development when designing UIs, how interacting with widgets work since they cannot just be attached to the screen like in PC. Attaching meshes to controller locations and rotate them accordingly to make it match where the hands are in real life. Then attach other components or actions like line traces to that location to properly setup user interaction. While building quality of life features, many small technical challenges that required specific knowledge, such as haptics or algorithms to efficiently show lines when pointing at something interactable and so on.

As said in Section 5, the showing of files in VR required creating a process that does not exist and has no solution, much less a correct one. This process was developed from testing other approaches such as converting files on upload, create a large png that can be scrolled from or create a pdf viewer in unreal engine which would take too much time to build. To design a png to texture method in C++ required understanding of how unreal engine assets are designed and how to read raw bytes through memory to create the texture and input it into a

dynamic material built from a material with a texture parameter. This entire process required a wide domain knowledge of how firebase works, efficient API design in Node and trying to integrate firebase with the application directly. Converting pdf to pngs in node was also a challenging process because of dealing with third party library solutions to process pdfs. Different libraries had different problems from security vulnerabilities from old, deprecated dependencies, slow execution time, or required external, unsupported programs that could not be added to the environment. The original intention was to create a massive scrollable png of different pages and make a scrollable material in unreal engine. But, due to how pdf-to-png process worked by generating pngs for every single page, stitching the pngs into one large image would be too large. Instead, every single png needs to be sent individually, which required stronger error catching on managing requests on the VR Application side because of the increase in number of requests. Because of the request and response being lightweight, the process time of the request is extremely low and feels instant and successfully created the entire process.

For the learning management system, the design of the website and loading resources was challenging. Every page required loading firebase services, scripts and other dependencies. These had to be loaded in an async manner to increase concurrency and decrease time. Navigating the exact order and managing authentication states for firebase to access all the services. The learning management system is an open-ended problem which had multiple solutions. Picking the correct stack to create the entire system based on which benefits and downsides offered by each framework. In addition to selecting the frameworks, managing network issues such as security, performance, and configuration of requests. This required research, testing and building an understanding of how to make different moving pieces align to create a robust, fast and efficient system.

## 8.2 Analysis of the results and future extensions

### 8.2.1 Learning Management System

The learning management system is functional and extendable infrastructure that can have more features added to it. Currently, it is a complete full-stack website that functions but lacks features compared to other systems such as canvas, google classrooms, etc. The main benefit of this LMS is how it ties to the VR Classroom. The file uploading and downloading system, authorization, and database leave easy room for expansion. Features such as assignments, quizzes, groups, profiles, and more. The addition of third-party tools, calendar systems, and methods to contact classmates and teachers are all features this is lacking compared to other LMS. These features fall out of the scope of this FYP but would be an important addition to make it match other systems. If the LMS reaches feature parity with other popular systems and includes features such as a 3D avatar designer and other VR features, it could be better than other offerings. However, this is dependent on the use of the VR Application since the main benefits it offers over other LMS rely on it. Only if the VR Application is successful can this be successful. In the future, to scale up, if more custom services and design is needed, then a database and a complete backend can be set up instead of the API infrastructure currently used with firebase.

### 8.2.2 VR Application

The VR Application is currently ready for production but has room for improvement. The classroom is fully functioning and can be used in a real classroom setting with multiplayer added to it. The classroom can be improved in terms of visual quality and design, and the models and materials can be improved. The material design of the walls and models can be reworked to be better optimized for lower quality and better performance. A different art style, such as low-poly or an overhaul in visual quality to lead to a more unified style instead of off-the-shelf materials for the most part. Concerning the models, custom-built models/meshes would also be nice to increase the visual quality of the application. In terms of functionality, fine-tuned and more quality-of-life upgrades, such as feedback on actions from sound or haptics or cues, would improve the user experience. On the other side, UI design can also be improved in terms of the design of the buttons, hover effects, images, and more. Lastly, the VR

Application could also be improved with some interactable tools or objects in class to use for class-related purposes such as drawing in 3D or more. The VR Class could also be extended with accessories such as VR pens or other tools to make it easier to write in VR. Additionally, a system for desk recognition or AR tools can also be added to enhance the experience. Hand tracking is also possible and has not been used in this project but can be leveraged in endless ways. On the rendering side, a system to dynamically adjust the refresh rate based on current performance would be a good improvement as well. The VR Classroom is a basic application right now with strong foundations, but there is no end to the potential for improvement and use cases it can be applied. The application is built in an extendable way to make improvements and integrations easier. To this end, a general interactable object class has been created with different actions for hover, interact, grab, and other actions. The entire application code is split into separate sections, such as the firebase subsystem, which is created to hold all the request-related code and functions. Different systems are built separately and compartmentalized into their own classes that can be built upon future expansion.

### 8.3 Achievements

#### Learning Management System

- Created robust, efficient, and extendable infrastructure.
- Stress and performance tested the system to accommodate more than hundreds of users for most actions.

#### VR Application

- Created a novel method of converting pdfs to pngs to show in classes
- Created production-ready, stress-tested VR application.
- Built in an extendable way to make future upgradability possible.

#### Overall

- Researched the market and came up with a niche to target.
- Planned, designed and executed an end-to-end system and complete feature set
- Overcame technical difficulties and lack of documentation and work to build from

## 8.4 Learning experiences

This project required an immense amount of planning and consideration and contained many technical challenges. While working on a large-scale project with multiple moving parts, planning, and research was crucial to its success. Many times, researching for alternate ways, documentation, and other examples helped devise new solutions. The development process had many technical challenges that, at points, seemed impossible, but through continuous effort in trying out different approaches, it was possible. Many times, documentation was not clear or available, and I had to manually bridge the gap between different software or figure out bugs in the official libraries and more. This is especially true for the file downloading feature or while designing the LMS infrastructure while being plagued with networking problems. There was a considerable amount of personal growth in terms of technical skills, planning, and attitude toward large projects.

## References

- Boletsis, C., & Chasanidou, D. (2022). *A typology of virtual reality locomotion techniques*. *Multimodal Technologies and Interaction*, 6(9), 72. <https://doi.org/10.3390/mti6090072>
- Bouchrika, I. (2022, October 11). *5 top college trends on LMS use by universities*. Research.com. Retrieved October 23, 2022, from <https://research.com/universities-colleges/college-trends-on-lms-use-by-universities>
- Browne, R. (2021, April 14). *Fortnite creator epic games' valuation jumps to \$29 billion in new funding round*. CNBC. Retrieved November 16, 2022, from <https://www.cnbc.com/2021/04/13/fortnite-creator-epic-games-valuation-jumps-to-29-billion.html>
- Class VR. (2017) *Virtual Reality for Schools*. ClassVR. Retrieved October 23, 2022, from <https://www.classvr.com/>
- Epic Games. (2019, June 20). *"Vader immortal" virtual reality experience: Face-to-face with vader himself*. Unreal Engine. Retrieved November 16, 2022, from <https://www.unrealengine.com/en-US/spotlights/vader-immortal-virtual-reality-experience-face-to-face-with-vader-himself>
- Parger, M. (2018). *Inverse Kinematics for Virtual Reality (thesis)*. Graz University of Technology, Graz. Retrieved from <https://diglib.tugraz.at/download.php?id=5c4a48dc5a282&location=browse>.
- IDC. (2022, June 30). *Meta's dominance in the VR market will be challenged in the coming years, according to IDC*. IDC. Retrieved November 14, 2022, from <https://www.idc.com/getdoc.jsp?containerId=prUS49422922>
- IrisVR. (n.d.). *The importance of frame rates*. IrisVR. Retrieved November 13, 2022, from <https://help.irisvr.com/hc/en-us/articles/215884547-The-Importance-of-Frame-Rates>

- Fordham University. (2020). *Types of online learning*. Fordham University: The Jesuit University of New York. Retrieved October 23, 2022, from [https://www.fordham.edu/info/24884/online\\_learning/7897/types\\_of\\_online\\_learning](https://www.fordham.edu/info/24884/online_learning/7897/types_of_online_learning)
- Iqbal, M. (2022, June 30). *Zoom revenue and usage statistics (2022)*. Business of Apps. Retrieved October 23, 2022, from <https://www.businessofapps.com/data/zoom-statistics/>
- Klemens, B. (2022, January 6). *I spent hundreds of hours working in VR. here's what I learned*. Wired. Retrieved October 23, 2022, from <https://www.wired.com/story/hours-working-VR-tips/>
- Lenovo. (2020). *VR Classroom*. Lenovo Tech Today US. Retrieved October 23, 2022, from <https://techtoday.lenovo.com/us/en/solutions/VR-classroom>
- Lemelin, C. (2021, October 7). *Why we don't turn our web cameras on in zoom: The impact on teaching and learning*. The Quad. Retrieved November 13, 2022, from <https://www.ualberta.ca/the-quad/2021/10/why-we-dont-turn-our-web-cameras-on-in-zoom-the-impact-on-teaching-and-learning.html>
- Matney, L. (2019, May 8). *How beat Saber beat the odds*. TechCrunch. Retrieved November 15, 2022, from <https://techcrunch.com/2019/05/08/how-beat-saber-beat-the-odds/>
- Miller, G. A. (1956). *The magical number seven, plus or minus two: Some limits on our capacity for processing information*. *Psychological Review*, 63(2), 81–97. <https://doi.org/10.1037/h0043158>
- Meta Quest. (n.d.). *Using Fixed Foveated Rendering*. Meta Quest for Developers. Retrieved March 28, 2023, from [https://developer.oculus.com/documentation/unity/unity-fixed-foveated-rendering/?locale=en\\_GB](https://developer.oculus.com/documentation/unity/unity-fixed-foveated-rendering/?locale=en_GB)

- Miller, K. (2022, September 26). *The benefits of online learning: 7 advantages of online degrees*. Northeastern University Graduate Programs. Retrieved October 23, 2022, from <https://www.northeastern.edu/graduate/blog/benefits-of-online-learning/>
- Ramachandran, V. (2021, February 23). *Four causes for 'zoom fatigue' and their solutions*. Stanford News. Retrieved October 23, 2022, from <https://news.stanford.edu/2021/02/23/four-causes-zoom-fatigue-solutions/>
- Shockley, K. M., Gabriel, A. S., Robertson, D., Rosen, C. C., Chawla, N., Ganster, M. L., & Ezerins, M. E. (2021). The fatiguing effects of camera use in virtual meetings: A within-person field experiment. *Journal of Applied Psychology*, 106(8), 1137–1155.  
<https://doi.org/10.1037/apl0000948>
- Soegaard, M. (2020). *Hick's law: Making the choice easier for users*. *The Interaction Design Foundation*. Retrieved November 16, 2022, from <https://www.interaction-design.org/literature/article/hick-s-law-making-the-choice-easier-for-users>
- Statista. (2022, October). *AR & VR - worldwide: Statista market forecast*. Statista. Retrieved October 23, 2022, from <https://www.statista.com/outlook/amo/ar-VR/worldwide>
- Unity Awards. (2018). *Unity awards 2018*. Unity Awards 2018. Retrieved November 15, 2022, from <https://awards.unity.com/2018>
- Wong, R. (2020). *When no one can go to school: Does online learning meet students' basic learning needs?* *Interactive Learning Environments*, 1–17.  
<https://doi.org/10.1080/10494820.2020.1789672>

## Appendix: A Monthly Logs

### A.1 March

Focused on performance and quality testing in VR and LMS systems. Dealt with issues in rendering and exporting and learned a deeper understanding of rendering in VR. Added small features in VR classroom that were on to do list if there was extra time.

Worked on the final report.

### A.2 February

- Focused on finishing the VR classroom
- Created system of showing files in VR
- Finished most of final touches of VR Classroom

### A.3 January

- Created a node server to act as in between the VR app and database.
- Setup access to database and VR communication

### A.4 December

- Created basic website, authentication, and authorization process.
- Created database and file management system.
- Managed hosting and setup

### A.5 November

Not much could be accomplished due to coursework. The focus was on seeing different rendering techniques, research, and optimization of VR performance.

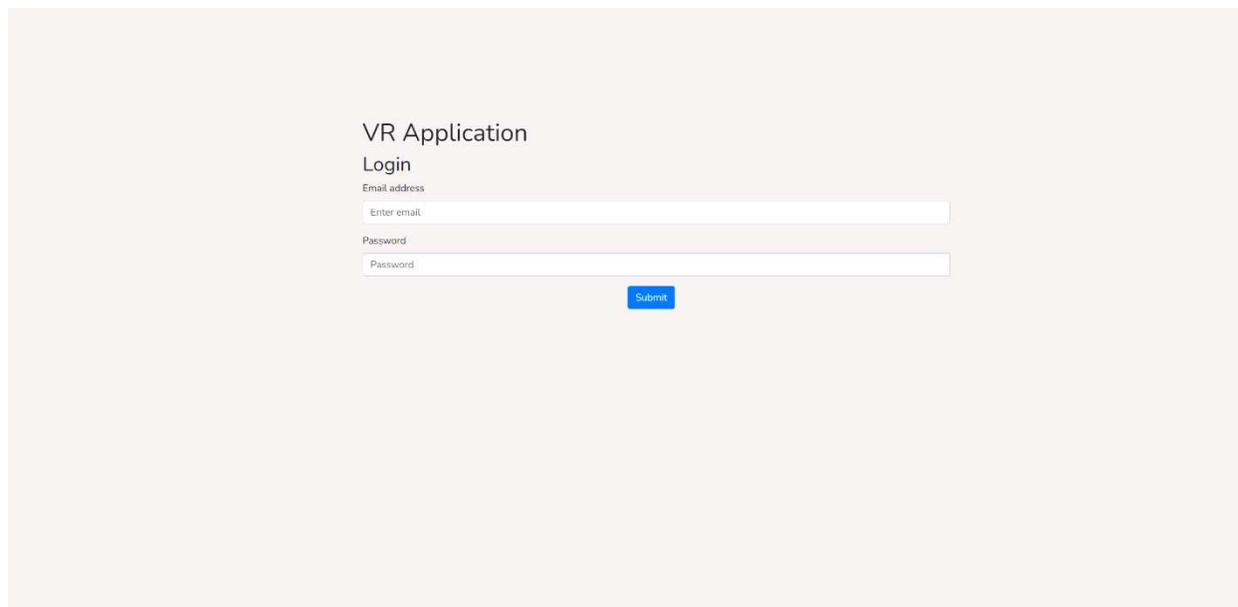
### A.6 October

This month was focused on developing a working version of the VR application. I started developing the first version of a classroom using simple assets to quickly test my design. I

moved on to the basics of movement, adding some animations. Lastly, I focused on making a beta of the whiteboard tool. The tool will allow teachers to draw on the whiteboard with different colors, sizes and more. I worked on another feature for scrolling through pdfs, as a first version it still has improvements in setting up the pipeline of how pdfs will become jpgs and integrate with the website. I also setup up a workflow for packaging and running the application purely on the headset. The interfacing between android SDK, unreal engine, visual studio, and android studio has been set up and a pipeline for future deployment.

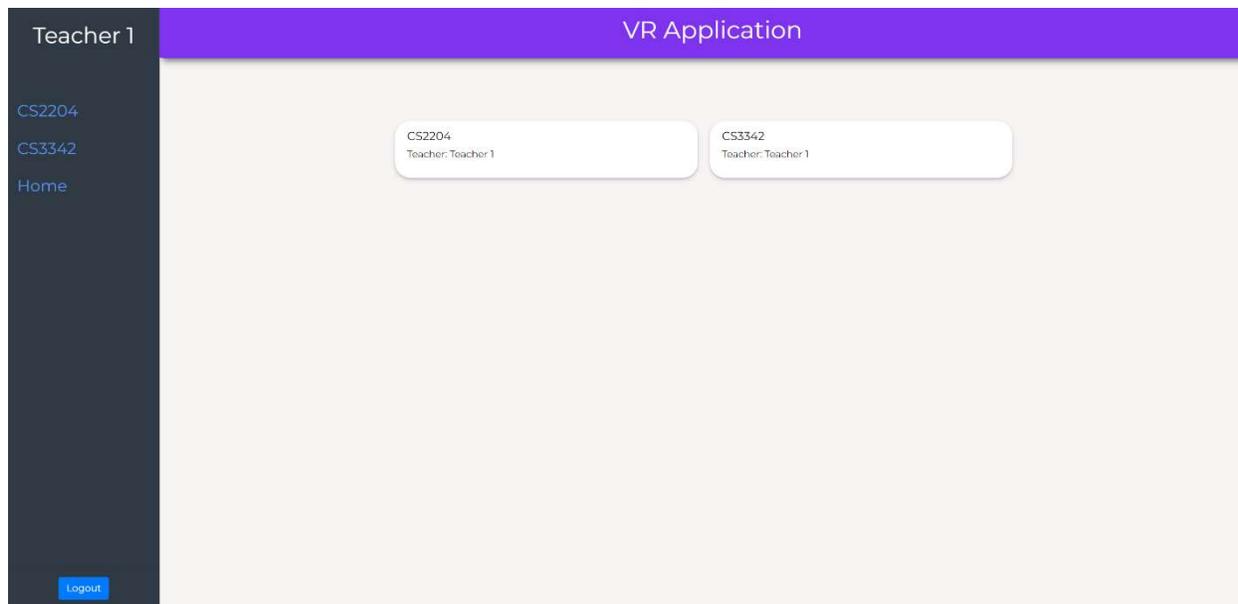
## Appendix: B Screenshots of Website

Figure B-1. Screenshot of the login page



The screenshot shows a login page for a VR Application. The page has a light beige background. At the top center, the text "VR Application" is displayed. Below it, the word "Login" is centered. Under "Login", there are two input fields: "Email address" with a placeholder "Enter email" and "Password" with a placeholder "Password". A blue "Submit" button is positioned below the password field.

Figure B-2. Screenshot of Home page; note username is 'Teacher 1'



The screenshot shows the home page of the VR Application. The page has a dark blue header bar with "VR Application" in white text. On the left side, there is a dark blue sidebar with the text "Teacher 1" at the top, followed by "CS2204", "CS3342", and "Home" in light blue. At the bottom of the sidebar is a blue "Logout" button. The main content area has a light beige background and contains two white rounded rectangular buttons. The left button displays "CS2204" and "Teacher: Teacher 1". The right button displays "CS3342" and "Teacher: Teacher 1".

Figure B-3. Screenshot of Class Page for a Teacher

The screenshot shows a class page for a teacher. The header is purple with 'Teacher 1' on the left and 'CS3342' on the right. The sidebar on the left contains links for 'CS2204', 'CS3342', 'Home', and a 'Logout' button. The main content area features an 'Upload file' section with a 'Choose File' button and an 'Upload file' button. Below this is a table of uploaded files.

| Filename   | Upload time       | Size (MB) | Delete |
|--|-------------------|-----------|--------|
| HW_1_1().pdf   | March 6, 2023     | 6.20      | Delete |
| Privacy_and_Ethics___GE1356_SDSC1001_Introduction_to_Data_Science_().pdf | March 24, 2023    | 0.22      | Delete |
| vr.pdf   | February 13, 2023 | 4.66      | Delete |
| GE150L_Chinese_Civilisation_4_Origins_and_Special_Traits.pdf             | February 11, 2023 | 11.47     | Delete |

Figure B-4. Screenshot of Class Page for a Student

The screenshot shows a class page for a student. The header is purple with 'Student 1' on the left and 'CS3342' on the right. The sidebar on the left contains links for 'CS2204', 'CS3103', 'CS3342', 'Home', and a 'Logout' button. The main content area features a table of uploaded files.

| Filename   | Upload time       | Size (MB) |
|--|-------------------|-----------|
| HW_1_1().pdf   | March 6, 2023     | 6.20      |
| vr.pdf   | February 13, 2023 | 4.66      |
| GE150L_Chinese_Civilisation_4_Origins_and_Special_Traits.pdf             | February 11, 2023 | 11.47     |
| Privacy_and_Ethics___GE1356_SDSC1001_Introduction_to_Data_Science_().pdf | March 24, 2023    | 0.22      |

## Appendix: C Screenshots of Final VR Classroom

Figure C-1. In-application screenshot of VR classroom with a file open. Note brightness and other visuals may look different than in VR.



Figure C-2. In-application screenshot of VR classroom while sitting down. Note brightness and other visuals may look different than in VR.

