GETO VR WORKSPACE

A PROJECT REPORT

Submitted by

KISHORE MS [211420104134]

LAXMAN MAHADEVAN R [211420104144]

PRADEEP K [211420104197]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

BONAFIDE CERTIFICATE

Certified that this project report "**GETO VR WORKSPACE**" the bonafide work of **KISHORE MS (211420104134), LAXMAN MAHADEVAN R (211420104144), PRADEEP K (211420104197)** who carried out the project work under my supervision.

SIGNATURE

Dr.L.JABASHEELA,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

Dr.S.HARIHARAN,M.E,M.TECH,Ph.D.,
ASSISTANT PROFESSOR
DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above mentioned students were examined in End Semester Project Viva- Voice held on.....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **KISHORE MS (211420104134), LAXMAN MAHADEVAN R (211420104144), PRADEEP K (211420104197)** hereby declare that this project report titled "**GETO VR WORKSPACE**", under the guidance of Dr.S.HARIHARAN is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

Name of the students

KISHORE MS (211420104134)

LAXMAN MAHADEVAN R (211420104144)

PRADEEP K (211420104197)

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express my sincere thanks to our beloved Directors **Tmt. C.VIJAYARAJESWARI**, **Dr. C.SAKTHI KUMAR, M.E., Ph.D** and **Dr. SARANYASREE SAKTHI KUMAR B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr. K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L.JABASHEELA, M.E., Ph.D.,** for the support extended throughout the project.

We would like to express our sincere thanks to **Project Coordinator Dr. G. SENTHIL KUMAR M.C.A., M.Phil.,M.B.A.,Ph.D.** and **Project Guide Dr. S. HARIHARAN,M.E, M.Tech, Ph.D** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

PROJECT COMPLETION CERTIFICATE

Spezar Tech Private Limited

Acknowledgment of Project Completion

For an Internship

22 March 2024

This is to certify that **Mr. PRADEEP K (211420104197)**, **Mr. KISHORE MS (211420104134)**, and **Mr. LAXMAN MAHADEVAN R (211420104144)**, students of final year **B.E. (COMPUTER SCIENCE ENGINEERING)** from **" PANIMALAR ENGINEERING COLLEGE (AUTONOMOUS) "** have successfully completed the Internship project titled **" GETO-VR WORKSPACE "** under the technology domain of **" VIRTUAL REALITY AND METAVERSE "**. The Internship project duration spanned from **January 2024** to **March 2024.**

During the project duration, they demonstrated commendable dedication, technical acumen, and teamwork in researching, developing, and implementing innovative solutions in the field of Virtual Reality (VR) and Metaverse technologies.

Through this Internship project, they have exhibited a profound understanding of advanced technological concepts, problem-solving skills, and the ability to work effectively in a collaborative environment. Their commitment to academic excellence and their enthusiasm for exploring and innovating within the rapidly evolving landscape of VR and Metaverse technologies are highly commendable.

Riyasdeen Samsudeen
Founder - CEO, Spezar Tech

Spezar Tech Private Limited
T.S.NO 253 South Kuruvikara Street, Pookulam
Thanjavur, Tamil Nadu, India, 613002
CIN : U72900TN2021PTC141102

+91 8610804852
info@spezartech.com

ABSTRACT

As interpersonal communication evolved over time, face-to-face meetings for significant business discussions were gradually replaced by technologically advanced alternatives like calls, texts, and video conferencing. The focus on video conferencing led to the development of standalone applications dedicated to this purpose, becoming the primary mode of communication for many companies. However, despite the effectiveness of conveying information, some individuals began to sense a detachment from the traditional office setting, particularly heightened during the pandemic. This realization spurred the emergence of immersive technologies, notably virtual reality (VR). VR provides a solution by allowing individuals to virtually experience specific locations or situations, aiming to bridge the gap between remote work and the physical office environment. The goal of this project is to enable workers to remain at home while still immersing themselves in the office atmosphere, fostering a sense of connection and productivity from any location they choose. This innovative approach of ours seeks to enhance the remote work experience and address the challenges posed by the changing landscape of communication and workplace dynamics. Through various available framework, this project helps one to communicate in a more effective and seamless manner with some interactivity features that enables to realize that employees are actually working in a cooperative environment with same level of focus and dedication as in the traditional way.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	vi
	LIST OF TABLES	ix
	LIST OF FIGURES	x
	LIST OF ABBREVIATIONS	xi
1.	**INTRODUCTION**	1
	1.1 Overview	2
	1.2 Problem Definition	2
	1.3 Proposed System	3
	1.4 Existing System	3
	1.5 Advantages and Disadvantages	3
2.	**LITERATURE REVIEW**	5
3.	**THEORETICAL BACKGROUND**	11
	3.1 Implementation Environment	11
	3.1.1 Hardware Requirements	12
	3.1.2 Software Requirements	12
	3.2 System Architecture	13
	3.2.1 Model Creation	13
	3.2.2 Server Communication	14
	3.2.3 Development Platform	16
	3.3 Proposed Methodology	17
	3.3.1 UML Diagrams	17
	a) Use Case Diagram	17
	b) Class Diagram	18
	c) State Diagram	19
	d) Activity Diagram	20
	e) Deployment Diagram	21

4.	**SYSTEM IMPLEMENTATION**	22
	4.1 User Authentication	23
	4.2 Player Profile	25
	4.3 Understanding the Environment	26
5.	**RESULTS AND DISCUSSIONS**	28
	5.1 Unit Testing	29
6.	**CONCLUSION AND FUTURE SCOPE**	30
	APPENDICES	32
	A.1 SDG Goals	32
	A.2 Source Code	33
	A.3 Screen Shots	54
	A.4 Plagiarism Report	56
	A.5 Paper Publication	57
	REFERENCES	58

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
5.1.1	Unit Testing	29

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.2.1	System Architecture	13
3.3.1.1	Use Case Diagram	17
3.3.1.2	Class Diagram	18
3.3.1.3	State Diagram	19
3.3.1.4	Activity Diagram	20
3.3.1.5	Deployment Diagram	21
4.1.1	Registration Page	23
4.1.2	Login Page	24
4.2.1	Profile Page	25
4.3.1	Environment	26
4.3.2	Keyboard	27
4.3.3	Creating or Joining a Room	27
7.1	3D Object	54
7.2	White Board	55
7.3	Ray Tracing	55
7.4	PDF importing	55

LIST OF ABBREVIATIONS

XR	Extended Reality
VR	Virtual Reality
AR	Augmented Reality
SDK	Software Development Kit
RAM	Random Access Memory
CPU	Central Processing Unit
SSD	Solid-State Drive
GB	Giga Byte
Mbps	Megabytes per second
LTS	Long Term Service

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 Overview

Communication is a key concept behind every successful organization. It helps us to better understand others and how an organization. Virtual Reality (VR) has become and important tool as technology improves to reach its peak. The concept of virtual reality enables users to interact with a three-dimensional environment through a computer-generated simulation. It can be used in a variety of fields, including medicine, engineering, and entertainment. One of the most common and widely used purpose of VR is for meeting and interacting with people in the computerized simulation. In this paper, we examine the concept of a conference meeting through the use of virtual reality. According to Natalie Massenet, “Always go into meetings or negotiations with a positive attitude.”. This system brings a positivity among users through the use of the immersive VR technology.

1.2 Problem Definition

In today's interconnected world, the day-to-day video conferencing tools fall short in delivering immersive remote collaboration experiences. These platforms often lack the spatial presence and interactivity necessary for effective communication in professional settings, such as business meetings and educational seminars.

Moreover, the static nature of traditional mechanisms fails to exhibit the dynamic and interactive nature of face-to-face interactions, which leads to the feeling of disconnection between users.

1.3 Proposed System

The aim of the project is to improve and replace the traditional approaches of online meeting tools. The proposed system utilizes Oculus technology to create an immersive VR conference room experience. With the help of servers and tools, users can experience the actual conference environment while staying in their desired place.

1.4 Existing System

1.4.1 VRChat

VRChat, a virtual world, invites you to roam, with avatars diverse, you're never alone. From bustling cities to whimsical lands, creativity flourishes, crafted by hands. In this digital realm, friendships ignite. As users converse through day and night. Games and activities, a plethora to explore, with uprises aplenty, there's always more. Events and gatherings, where stories unfold, in VRChat's universe, adventures untold. A haven for connection, where dreams take flight, VRChat beckons all to its immersive delight.

1.4.2 AltspaceVR

AltspaceVR revolutionizes social interaction by providing a virtual reality platform where users can connect, communicate, and collaborate in immersive environments. Through customizable avatars and virtual spaces, individuals can express themselves and engage with others authentically. The platform's emphasis on social events, ranging from concerts to workshops, fosters shared experiences and meaningful connections. With cross-platform compatibility, AltspaceVR ensures accessibility across various VR devices and even allows participation via PC or Mac without a headset. Users can create their own virtual spaces, tailoring environments for personal or professional use.

1.5 Advantages and Disadvantages

The GETO VR WORKSPACE using virtual reality (VR) technology in the aspect of conference meeting that has both advantages and disadvantages. One advantage is that VR technology allows for a more immersive and interactive experience, which can increase engagement and retention of information. Participants can experience the actual workplace so that they can maintain their pace of productivity and also improve

the quality of sharing information. Another advantage of utilizing VR technology in conference meetings is the potential for enhanced collaboration and teamwork. Participants can interact with virtual objects and environments in real-time, facilitating brainstorming sessions, problem-solving activities, and collaborative projects regardless of geographical location. This fosters a sense of presence and closeness among team members, leading to improved communication and productivity.

Furthermore, VR technology offers flexibility and accessibility, allowing participants to join meetings from anywhere with an internet connection. This eliminates the constraints of physical distance and travel, saving time and resources for both individuals and organizations. Additionally, VR can accommodate larger groups of participants compared to traditional meeting spaces, enabling organizations to host events with a wider audience and reach.

Despite these benefits, there are still some challenges associated with VR technology in the conference meeting context. For instance, ensuring equal access and inclusivity for all participants, including those with disabilities or limited access to technology, remains a concern. Additionally, maintaining data security and privacy in virtual environments requires robust systems and protocols to safeguard sensitive information.

In conclusion, while VR technology offers numerous advantages for enhancing conference meetings, including increased engagement, collaboration, and flexibility, it also presents challenges such as high initial costs, technical expertise requirements, and potential accessibility issues. By carefully addressing these challenges and leveraging the strengths of VR technology, organizations can unlock new possibilities for immersive and effective conference experiences.

CHAPTER 2

LITERATURE REVIEW

1. Title : Empowering Knowledge with Virtual and Augmented Reality

Year : 2023

Authors : Osvaldo Gervasi, Damiano Perri, Marco Simonetti

Abstract :

Recent global advancements in ICT technologies have motivated our research to assess the impact of Virtual and Augmented Reality on teaching and learning. In particular, we conducted a rigorous study on the effects on learning by subjecting classes of students to an experience of these emerging technologies and measuring their feelings before and after the experiment through questionnaires. We carried out the research involving three high schools and a total of 162 students aged between 15 and 20. During the meetings, we made available two applications that we developed. The first is a Virtual Reality environment that enables the user to explore a room in which the system spawns three-dimensional objects defined by the teacher. The objects may represent mathematical functions, physics simulations, scenes relating to historical re-enactments, etc. The second is an Augmented Reality smartphone application that allows three-dimensional figures to be observed via the smartphone screen by framing a marker printed on paper. Both applications are realised with free software, trying to minimize the technical requirements for their operation and guaranteeing reliability and usability. The work concludes by analysing the results obtained and comparing the effectiveness of the two technologies tested. Results of tests conducted in three Italian schools show that Augmented Reality is seen as a useful tool in education by about 94.43%. Importantly, the use of these technologies requires an inclusive design that involves all students, regardless of their level of familiarity with the technology. The overall analysis reveals that both technologies performed well, and students value them for their intuitiveness and level of immersion. The in-presence experience is more effective for using immersive applications fostered by student collaboration.

2. Title : The Metaverse: Applications, Concerns, Technical Challenges, Future Directions and Recommendations

Year : 2023

Authors : Osvaldo Gervasi, Damiano Perri, Marco Simonetti

Abstract :

The Metaverse is all about expanding connectivity amongst users and objects and seamlessly delivering information and services to the right user at the right time. Its potential advantages are virtually limitless, and its applications are progressively changing the way we live, and are opening new opportunities for innovation and growth. It is crystal clear that the Metaverse can enable fully immersive experience, elements of fantasy, and new degrees of freedom. However, it is still considered controversial since it will also open up opportunities for misconduct and crime. Furthermore, the industry lacks the capacity to carry out a comprehensive study of the potential risks that will come along. This paper highlights the current and envisioned Metaverse applications along with the main concerns and challenges faced by the Metaverse stakeholders. Furthermore, it examines the strengths, weakness, opportunities and threats of the Metaverse technology. Finally, the paper presents the future directions and highlights the most important recommendations for developing the Metaverse systems.

3. **Title :** A Review of Metaverse's Definitions, Architecture, Applications, Challenges, Issues, Solutions, and Future Trends

Year : 2022

Authors : Abbas M. Al-Ghaili, Hairoladenan Kasim, Naif Mohammed Al-Hada, Zainuddin Bin Hassan, Marini Othman, Jakir Hussain Tharik, Rafiziana Md. Kasmani, Ibraheem Shayea

Abstract :

Metaverse is a vision enabling to constitute an environment in which someone could see real and virtualized worlds. The Metaverse is a product (or something similar as we do not yet know its final form) of such technologies. In this circumstance, when applications that utilize the Metaverse are used, there seem to be no transportation charges, and there is no cap on amounts of individuals, users, players, learners, or trainee who can take part. Hence, and due to such a feature, the Metaverse has attracted various researchers from different fields where it has been exploited by them to contribute to those fields and research areas. As for example, it is possible to teach various target audiences by offering different events and classes from any location in the globe. In order for a participant to utilize the Metaverse, there are necessary conditions to be considered as well as other settings to be initialized. In line of this, virtual reality, augmented reality, availability of required sensors, smart glasses, headsets, and few others are considered some examples of such conditions and settings that the Metaverse requires. Despite the advantages that Metaverse offers us, there are a number of considerations that must be taken into account while developing it by interested researchers. One of these concerns is the Metaverse privacy regarding the participants (represented as avatars inside the Metaverse environment). Another issue is that since the Metaverse is still in its early stages, many attempts have to be made from interested researchers who engage to develop it to enhancing it. This review aims to survey related articles that concern the Metaverse and its development providing a review of the chronological stages throughout the history of the development of

Metaverse. It aims also to list a number of recent technological advances allowing the Metaverse. Besides, Metaverse's definitions, properties, architecture, and applications have been discussed and listed in this review. The novelty of this article is that it has suggested a framework to a number of issues that are still paid attention for potential solutions by researchers aiming to contribute to researchers and designers to consider such an issue and its corresponding solution for future research works and enhancement. Challenges faced by researchers and other relevant concerned issues related to Metaverse have been in detail discussed and highlighted. Besides, future trends have been clarified.

4. **Title :** Design of a Desktop Virtual Reality-Based Collaborative Activities Simulator (ViRCAS) to Support Teamwork in Workplace Settings for Autistic Adults

Year : 2023

Authors : Ashwaq Z. Amat, Deeskha Adiani, Mahrukh Tauseef, Michael Breen, Spencer Hunt, Amy R Swanson, Amy S. Weitlauf, Zachary E. Warren, Nilanjan Sarkar

Abstract :

Autistic adults possess many skills sought by employers, but may be at a disadvantage in the workplace if social-communication differences negatively impact teamwork. We present a novel collaborative virtual reality (VR)-based activities simulator, called ViRCAS, that allows autistic and neurotypical adults to work together in a shared virtual space, offering the chance to practice teamwork and assess progress. ViRCAS has three main contributions: 1) a new collaborative teamwork skills practice platform; 2) a stakeholder-driven collaborative task set with embedded collaboration strategies; and 3) a framework for multimodal data analysis to assess skills. Our feasibility study with 12 participant pairs showed preliminary acceptance of ViRCAS, a positive impact of the collaborative tasks on supported teamwork skills practice for autistic and neurotypical individuals, and promising potential to quantitatively assess collaboration through multimodal data analysis. The current work paves the way for longitudinal

studies that will assess whether the collaborative teamwork skill practice that ViRCAS provides also contributes towards improved task performance.

5. Title : Toward VR in VR: Assessing Engagement and Social Interaction in a Virtual Conference

Year : 2022

Authors : Catarina Moreira, Francisco P. M. Simoes, Mark J. W. Lee, Ezqruiel R. Zorzal, Robert W. Lindeman, Joao Madeiras Pereira, Kyle Johnsen, Joaquim Jorge

Abstract :

The pandemic brought about an unprecedented number of virtual conferences, given the heavy restrictions on travel to in-person meetings. Despite all the advances in technology, people still complain about virtual events. There is Zoom fatigue, confinement malaise, and a longing for personal social interactions. This paper discusses our experience organizing the IEEE Virtual Reality Conference (IEEE VR) as a virtual event. IEEE VR was a success with 1200+ registered paying participants, dozens of workshops and tutorials, and hundreds of technical papers. We used (1) a virtual environment platform, together with (2) discussion tools and (3) videoconferencing/broadcast/online tools to further provide effective social interaction and increase engagement. In this paper, we explore the synergies between virtual environments and other online tools and assess user engagement by analyzing the messages exchanged between participants across different genders and geographical regions. To this end, we apply diverse engagement metrics for online conferences. Our analysis shows that these metrics have the potential to highlight engagement, diversity, and inclusion by combining textual messages, participant geographic and gender information, communities of participants, and visitation patterns in a virtual environment. Drawing on our results and experiences, we propose guidelines for organizing technical virtual events to increase diversity.

CHAPTER 3
THEORETICAL BACKGROUND

3. THEORETICAL BACKGROUND

The following are the hardware and software requirements for the application:

3.1 Implementation Environment

3.1.1 Hardware Requirements

1) Personal Desktop

RAM	:	16 GB RAM
Graphics Card	:	Compatible 4 GB Graphics Card
CPU	:	Ryzen 5 or Intel i5 equivalent CPU or above

2) Internet Router

Download Speed	:	25
Mbps, Upload Speed	:	5 Mbps.
Router	:	5 Ghz WIFI Router.

3) Server Requirements

Account Access	:	Oculus, Unity Account Access.
Security Services	:	Access to local networks.

3.1.2 Software Requirements

The following is the SDK and software that are required to build the application:

1) SOFTWARES USED

- Unity 3D – version (2021.3.1f1 LTS)
- Blender 3D
- SketchUp
- Oculus Software
- SDK: XR Interaction Toolkit

2) PACKAGES USED

- Asset Store Originals
- XR Interaction Toolkit Package

- XR Plugin Manager
- Oculus XR
- Visual Studio

3.2 System Architecture

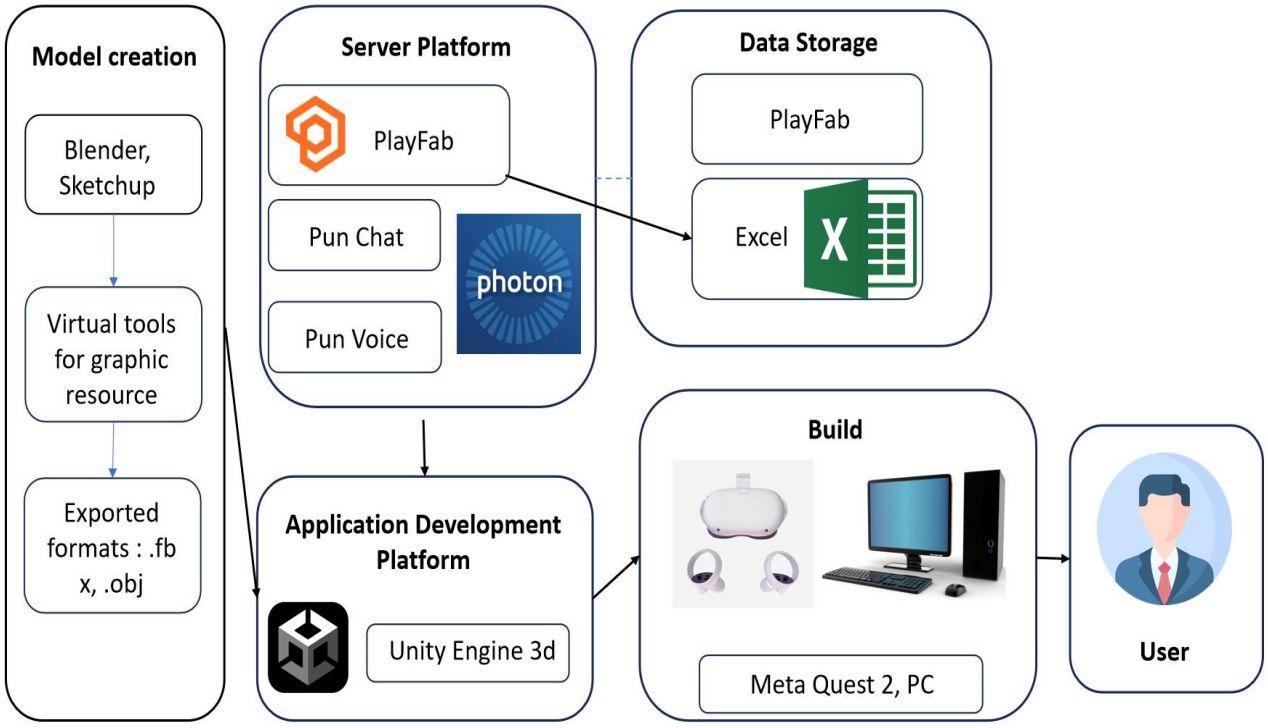


Figure 3.2.1 – System Architecture

3.2.1 MODEL CREATION

Creating a 3D model using tools like Blender or SketchUp involves a detailed and systematic process, beginning with conceptualization and design. In SketchUp, a user-friendly 3D modeling software, designers leverage its intuitive interface and powerful toolset to bring their ideas to life in three-dimensional space. The process often starts with rough sketches or reference images, serving as a foundation for the model. These sketches help outline the basic shapes and proportions of the objects or environments being created.

Once the initial concept is established, designers move on to the modeling stage, where they use SketchUp's array of modeling tools to construct the digital geometry of the

objects. This involves techniques such as extrusion, push/pull, and manipulation of edges and faces to refine the shape and form of the model. SketchUp's simplicity and flexibility allow for quick iterations and adjustments, making it ideal for rapid prototyping and concept development.

After the basic geometry is established, attention is turned to detailing and texturing. Designers utilize SketchUp's material library and painting tools to add surface textures, colors, and materials to the model. This step is crucial for enhancing realism and visual appeal. Textures can range from simple colors to intricate patterns and materials like wood, metal, or glass, applied to different parts of the model as per the design requirements.

Once the model is fully textured, it can undergo further enhancements such as adding components, furniture, or landscaping elements to complete the scene. SketchUp's extensive 3D Warehouse provides a vast collection of pre-made models that can be easily imported and incorporated into the design.

Finally, the model is ready for presentation or visualization. SketchUp offers various rendering plugins and extensions that allow designers to create photorealistic images and animations. Lighting, camera angles, and rendering settings can be adjusted to achieve the desired visual aesthetic. Rendering may take some time, depending on the complexity of the scene and the level of detail required.

3.2.2 SERVER COMMUNICATION

Unity's integration with third-party servers like Photon Unity Networking (PUN), Photon Realtime, and Microsoft PlayFab represents a significant advantage for developers seeking to create robust multiplayer experiences and manage online services effectively within their Unity projects.

Photon Unity Networking (PUN) is a popular solution for real-time multiplayer games, offering a high-level API and a user-friendly interface for Unity developers. PUN simplifies the process of setting up and managing multiplayer functionality by handling tasks such as server connections, room creation, and player synchronization. It provides features like remote procedure calls (RPCs) for communication between players, built-

in support for latency compensation, and flexible matchmaking options. Developers can leverage PUN's extensive documentation and community support to quickly integrate multiplayer capabilities into their games, whether they're creating small-scale indie projects or large-scale multiplayer experiences.

Similarly, Photon Realtime extends the capabilities of PUN by offering a lower-level API for developers who require more control over their networking infrastructure. With Photon Realtime, developers can create custom server logic, implement advanced networking features, and optimize performance for specific gameplay requirements. This flexibility makes Photon Realtime a preferred choice for developers working on complex multiplayer projects that demand precise control over networking protocols and server configurations.

On the other hand, Microsoft PlayFab serves as a comprehensive backend platform for managing online services such as player authentication, game telemetry, in-game economy, and more. PlayFab seamlessly integrates with Unity through its Unity SDK, allowing developers to leverage its powerful features without leaving the Unity Editor. With PlayFab, developers can easily implement features like player authentication using popular identity providers, manage player data and progression, create dynamic in-game events, and analyze player behavior through real-time analytics. PlayFab's scalability and reliability make it an ideal solution for developers looking to build and operate live games with millions of players, providing a robust infrastructure for managing online services and engaging player experiences.

Overall, Unity's integration with third-party servers like PUN, Photon Realtime, and Microsoft PlayFab empowers developers to create immersive multiplayer experiences and manage online services efficiently. Whether they're developing small-scale multiplayer games or large-scale live operations, these solutions offer the tools and infrastructure needed to bring their vision to life and deliver compelling gaming experiences to players worldwide.

3.2.3 DEVELOPMENT PLATFORM

Unity 3D engine stands as a powerhouse in the realm of game development, offering an expansive toolkit and robust capabilities for creators across various industries. Renowned for its accessibility and versatility, Unity empowers developers to craft immersive experiences for not only games but also applications in fields such as architecture, automotive, film, education, and beyond. With its intuitive interface and extensive asset store, Unity accelerates the development process, enabling creators to focus on innovation rather than grappling with complex coding challenges. Its cross-platform compatibility ensures that projects can reach a wide audience, spanning mobile devices, consoles, PCs, augmented reality (AR), virtual reality (VR), and mixed reality (MR) platforms. Unity's real-time rendering engine facilitates the creation of stunning visuals, while its physics engine provides realistic interactions and dynamics within virtual environments. Furthermore, Unity's comprehensive development environment supports collaborative workflows, allowing teams to work seamlessly together regardless of their location. Whether its indie developers bringing their creative visions to life or established studios pushing the boundaries of technology, Unity serves as a cornerstone for innovation and creativity in the ever-evolving landscape of digital experiences. Its uses extend far beyond gaming, empowering creators to build interactive simulations, training programs, architectural visualizations, medical applications, and more. With its vast community, extensive documentation, and continuous updates, Unity remains at the forefront of empowering creators to realize their dreams and push the boundaries of what's possible in interactive entertainment and beyond.

3.3 Proposed Methodology

3.3.1 UML Diagrams

a) Use Case Diagram

A use case describes how a user uses a system to accomplish a particular goal. A use case diagram consists of the system, the related use cases and actors and relates these to each other to visualize the system, actors and use case. Use cases help ensure that the correct system is developed by capturing the requirements from the user's point of view.

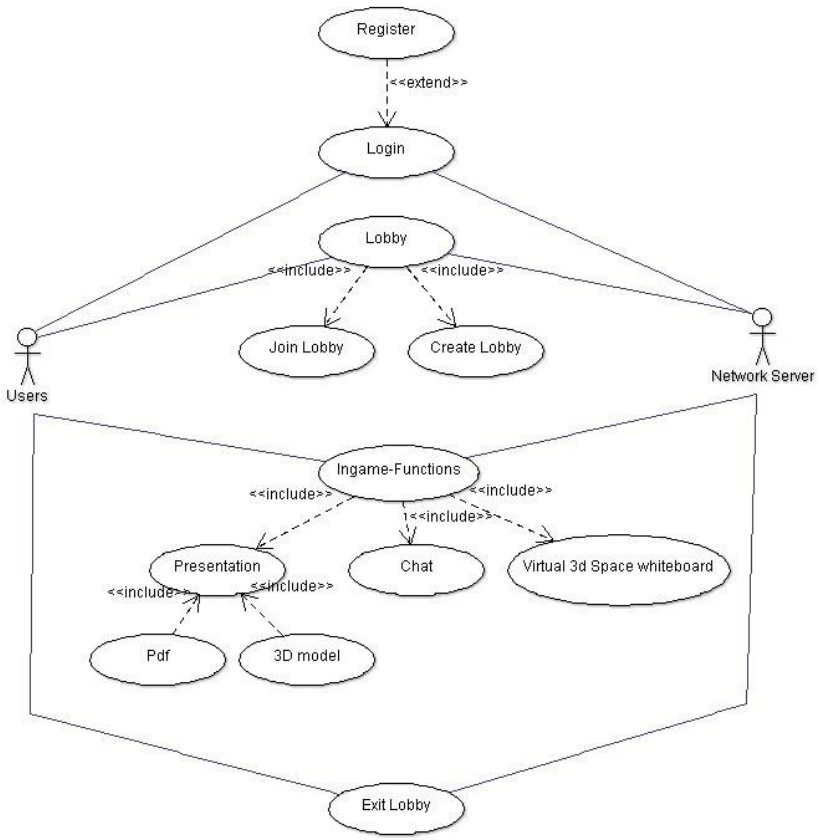


Figure 3.3.1.1 – Use Case Diagram

b) Class Diagram

A class diagram describes the structure of an object-oriented system by showing the classes in that system and the relationships between the classes. A class diagram also shows constraints, and attributes of classes. Class is represented as rectangular box showing class name, attributes, operations. An attribute is a logical data value of an object. Attributes of a classifier also called structural properties in the UML.

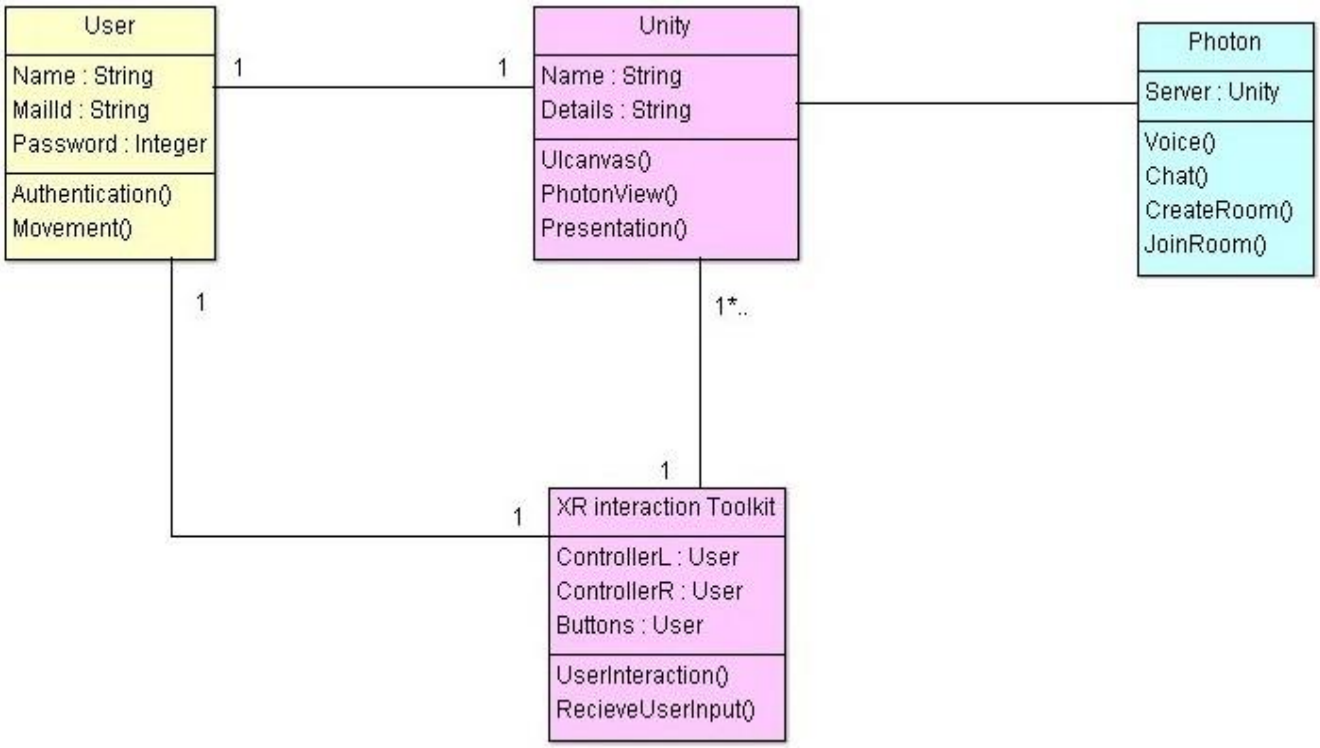


Figure 3.3.1.2 – Class Diagram

c) State Diagram

A state diagram, also known as a state machine diagram, illustrates the behavior of a system by depicting its various states and the transitions between them triggered by events. It consists of states represented as rounded rectangles, transitions depicted as arrows connecting states, and events triggering these transitions. The diagram typically starts with an initial state and ends with a final state, indicating the beginning and termination points of the system's behavior. State diagrams are crucial in software engineering for modeling complex systems, understanding their behavior, and designing finite state machines for applications like user interfaces and control logic.

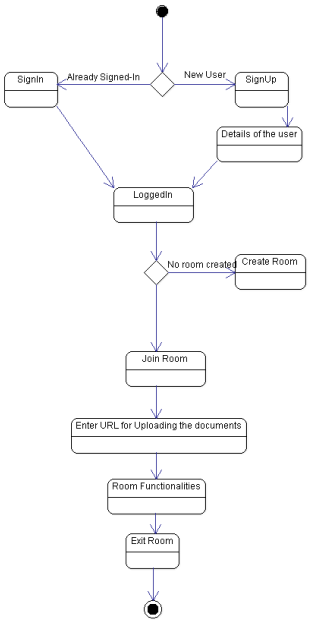


Figure 3.3.1.3 – State Diagram

d) Activity Diagram

Activity diagrams are graphical representations used in software engineering to depict the flow of activities or actions within a system or process. These diagrams utilize various symbols such as activities, actions, decision points, and control flow arrows to illustrate the sequence of tasks involved. Typically, they begin with an initial node marking the starting point and end with a final node indicating the termination of the process. Decision nodes enable branching based on conditions, while merge nodes consolidate multiple paths back into a single flow. Activity diagrams serve as valuable tools for understanding and communicating complex workflows in software development and business processes, aiding stakeholders in analyzing system behavior and ensuring clarity in requirements and design phases.

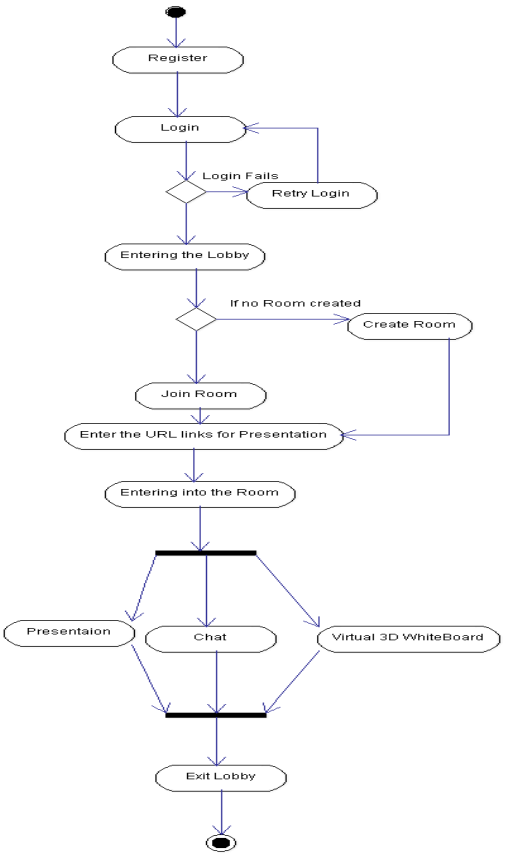


Figure 3.3.1.4 – Activity Diagram

e) Deployment Diagram

In the UML, deployment diagram is used to visualize the static aspect of these physical nodes and their relationships and to specify their details for construction. Deployment diagrams address the static deployment view of an architecture.

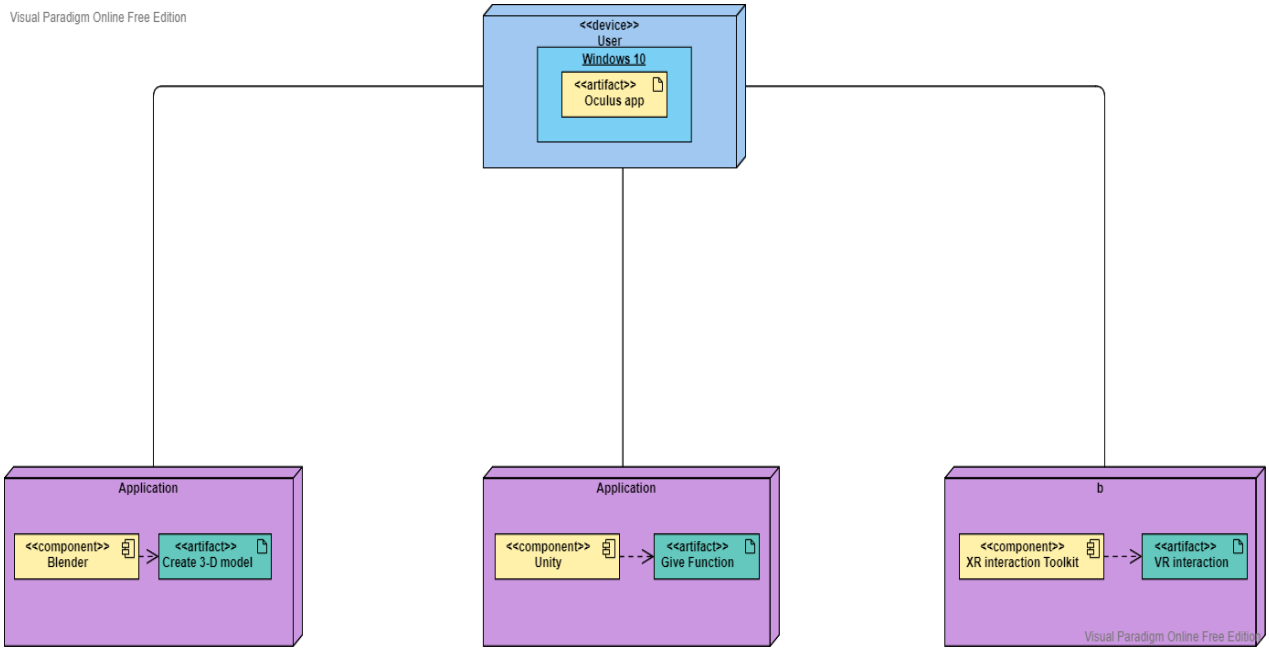


Figure 3.3.1.5 – Deployment Diagram

CHAPTER 4

SYSTEM IMPLEMENTATION

4. SYSTEM IMPLEMENTATION

This subsection contains the system implementation of this VR Software Application.

4.1 User Authentication

1) Registration Page

The registration page collects details such as username, email id, password for your account, your role in the organization and phone number.

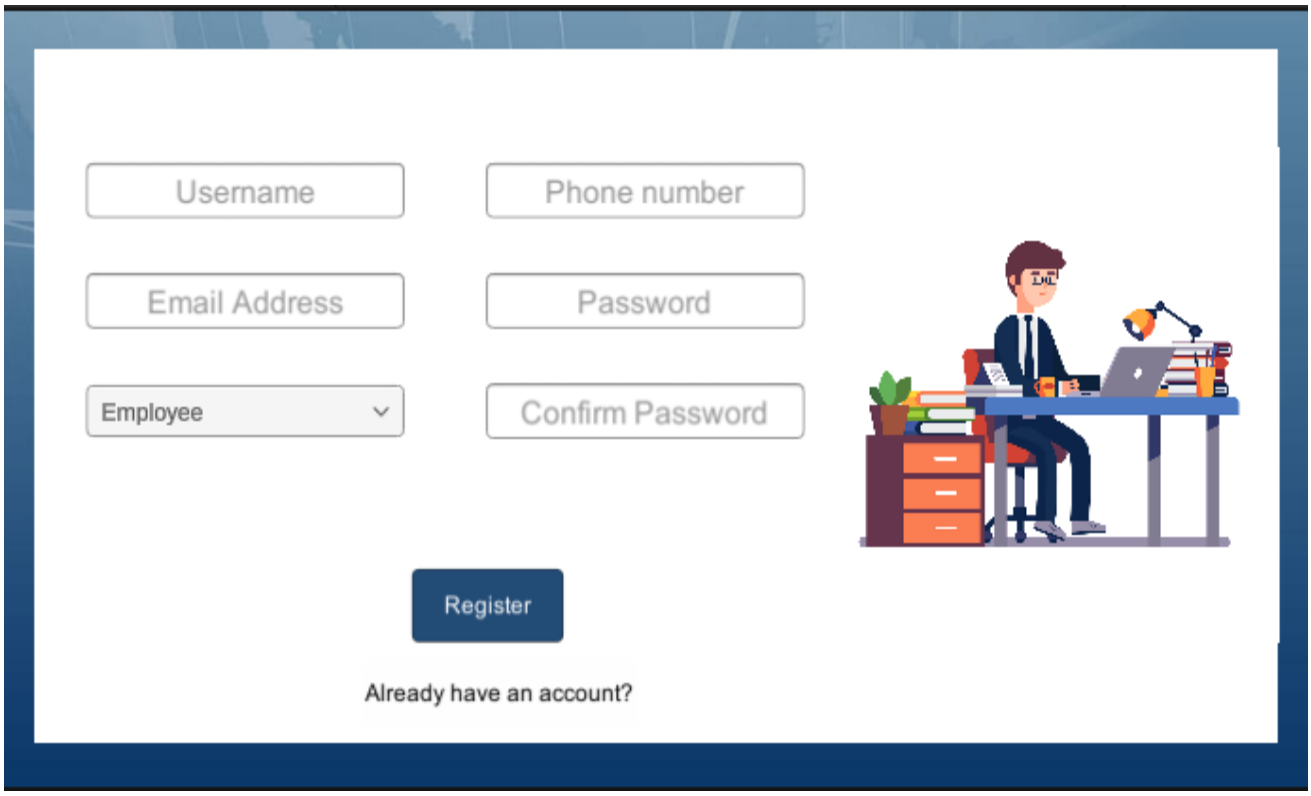


Figure 4.1.1- Registration Page

2) Login Page

The login page prompts the user to enter the registered email id and password to verify against the records stored in the Playfab cloud and allows the authorized users to enter into the lobby scene.

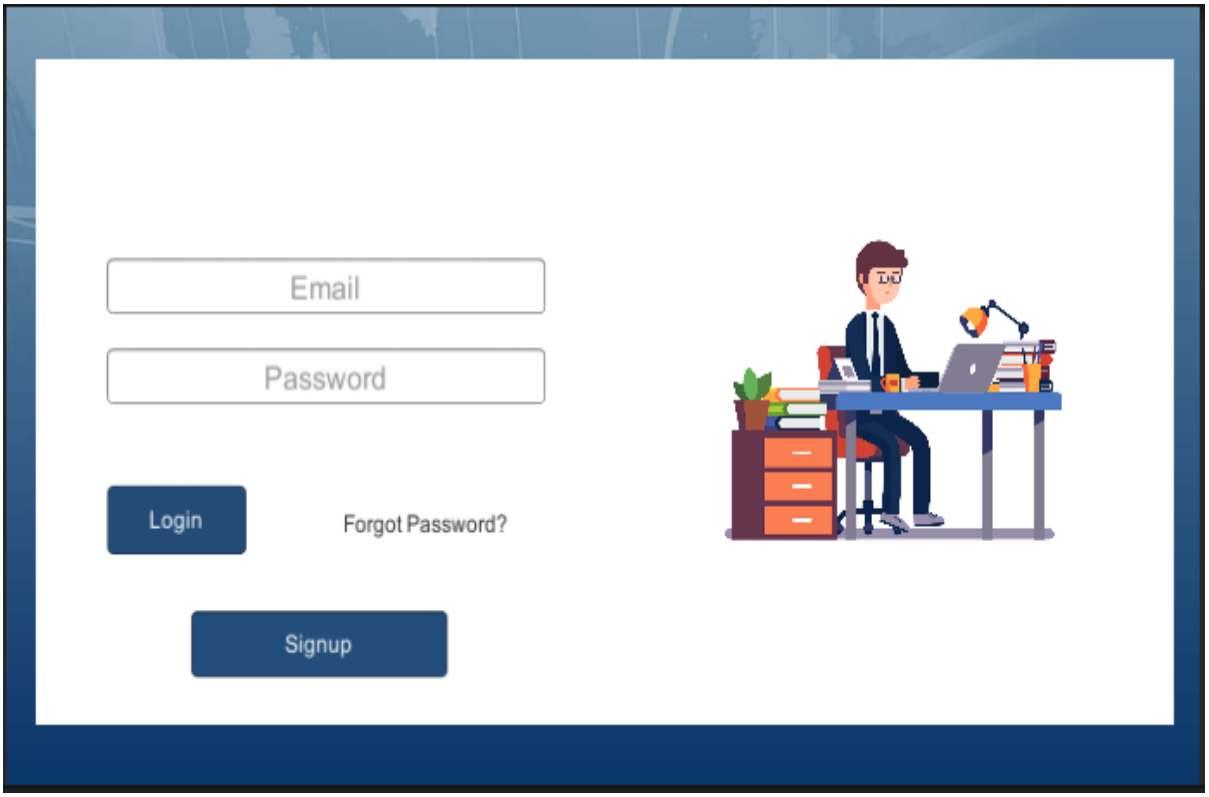


Figure 4.1.2- Login Page

4.2 Player Profile

The player profile page provides the general data of the current player logged in such as Name, Email, Gender, Phone number and Role of the son in the organization.

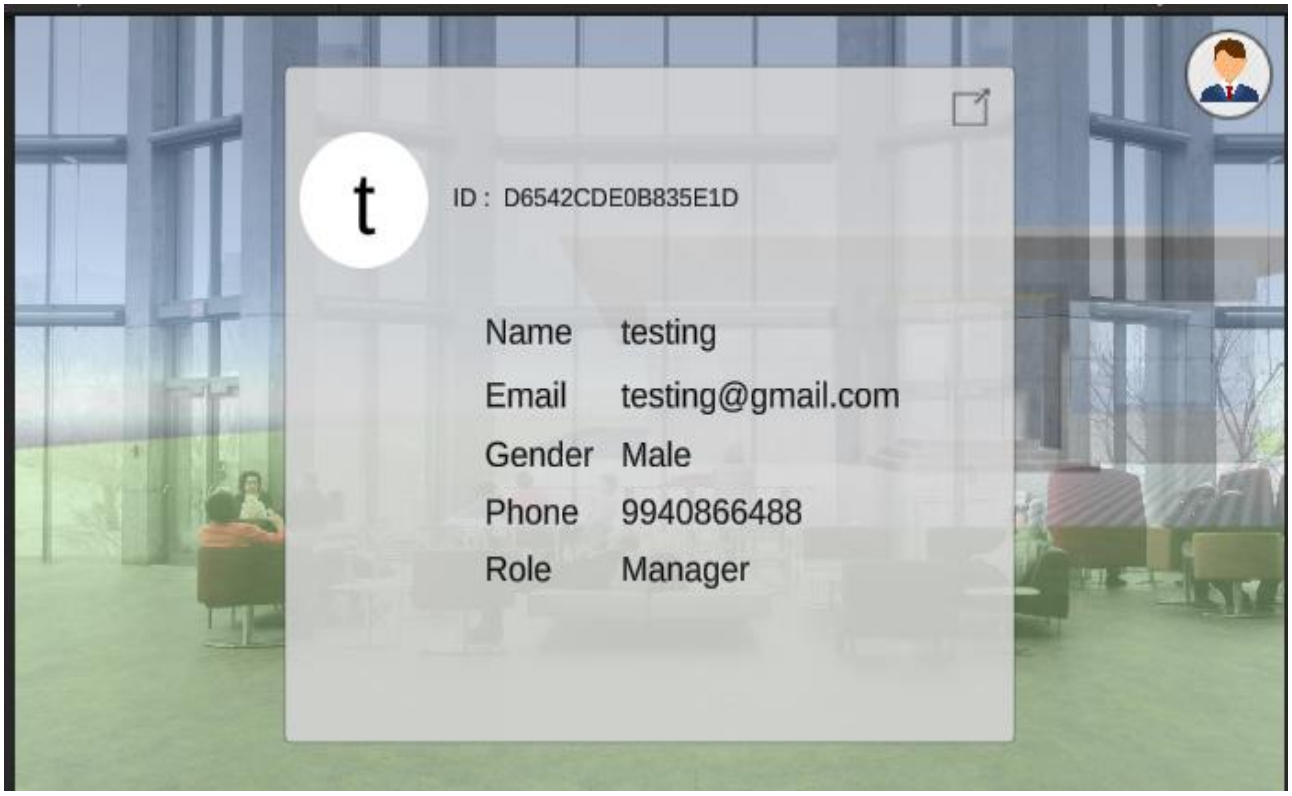


Figure 4.2.1- Profile Page

4.3 Understanding the Environment

1.Once the player logs in to their account, they can either create a room or join a room.

2.After the room is created by a user, others are able to join to the same room.

3.Users can chat among other users present within the same room. Both public and private messages are supported.

4.Users can communicate with others present in the room using the PUN voice plugin.

5.Once all users have entered the room, they can present pdf, write on whiteboard and also type on the virtual keyboard through the hand tracking technology of Oculus Quest 2.

6.Users can roam freely throughout the environment.

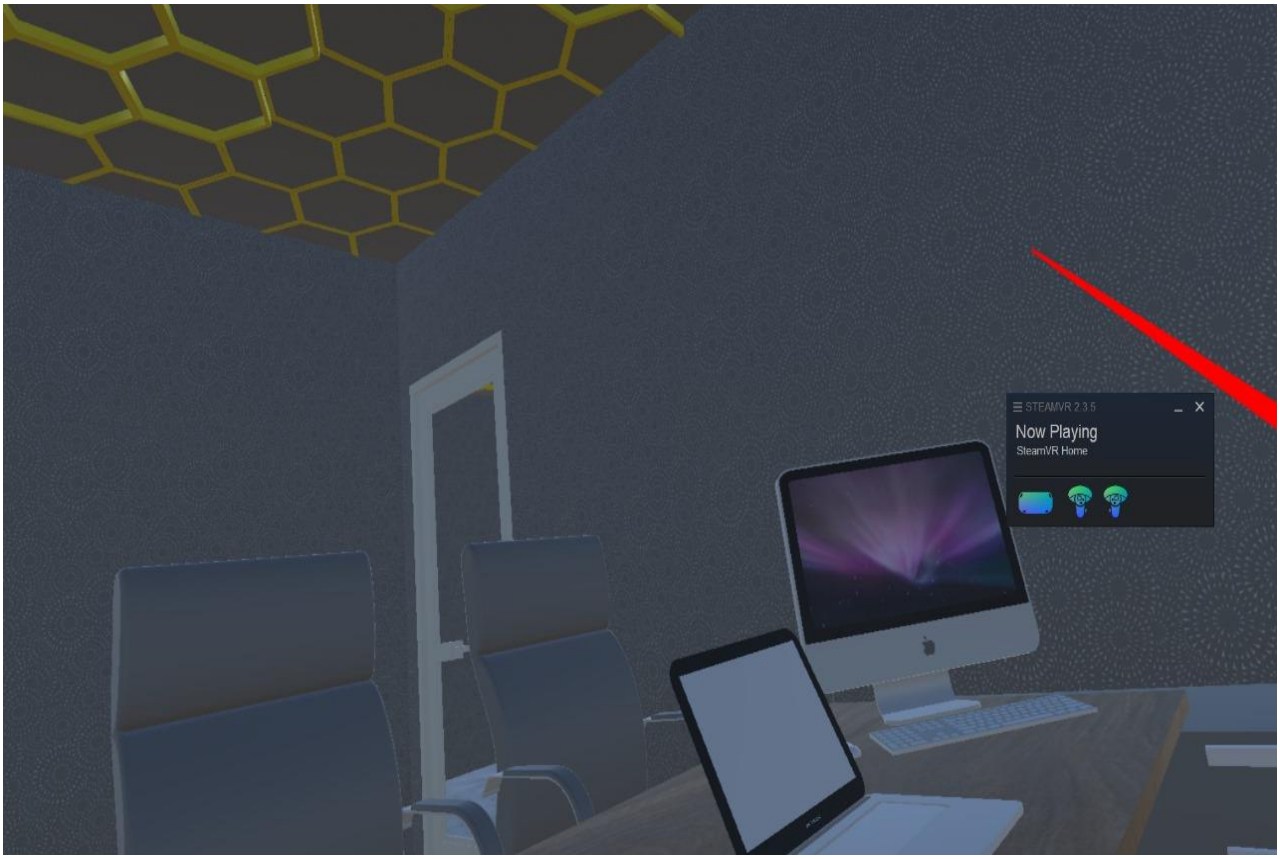


Figure 4.3.1- Environment

Figure 4.3.2- Keyboard

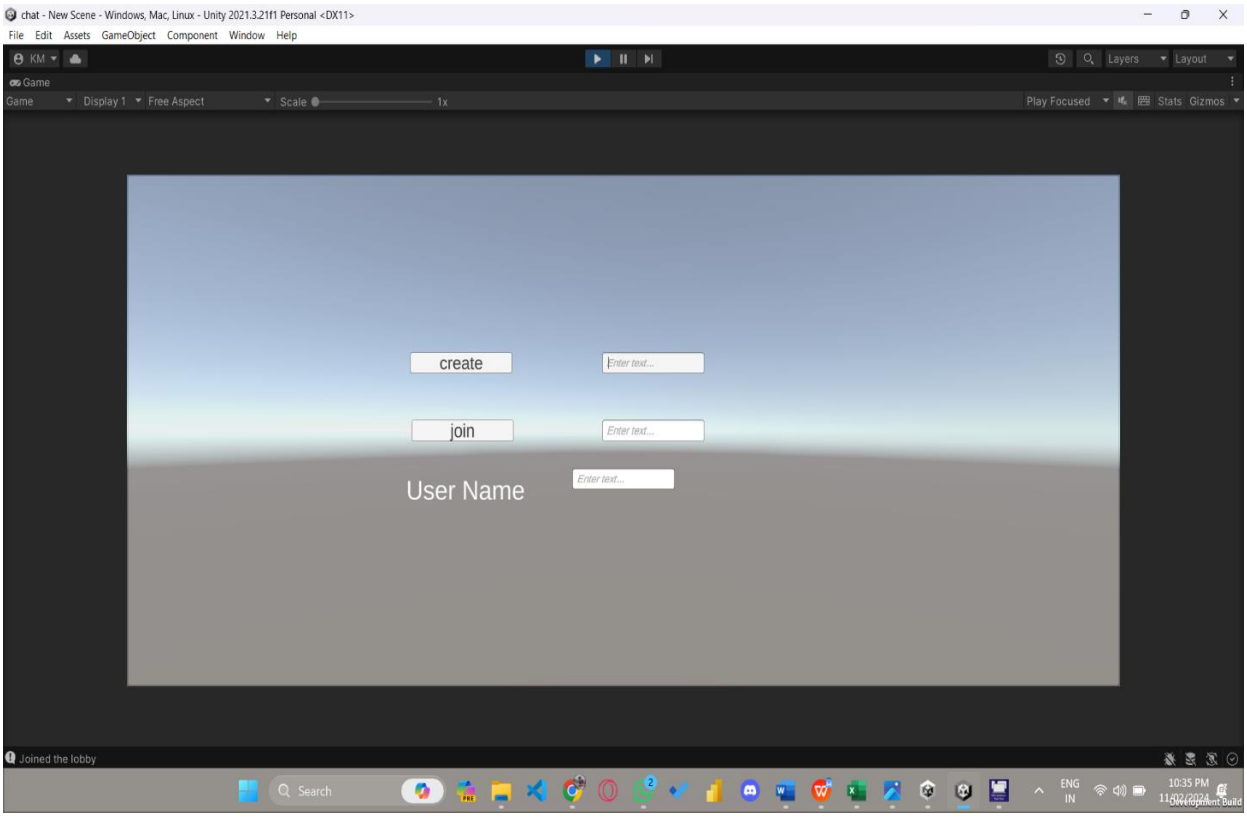


Figure 4.3.3- Creating or joining a room

CHAPTER 5

RESULTS AND DISCUSSIONS

5. TESTING

To keep the system error free during the phases of development and during the time when new features are added, the following testing strategies are applied:

5.1 Unit Testing

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

Each module can be tested using the following two Strategies:

Table 5.1.1 – Unit Testing

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Data	Expected Result	Actual Outcome	Status (Pass / Fail)
1	Authentication of users	Enter user registered email and password	The data entered must be present in the Playfab database	Email : testing@gmail.com Password : testing@123	Enter the lobby scene	player entered to the lobby scene	Pass
2	Joining a room	Trying to join a room that is not created	Room name that is to be entered in Join input field section must not be present in the pun server	"Test_Room"	Indications related to invalid room details	"The room you entered does not exist"	Pass
3	Chat	Send a private chat to a user without others notified	The user must be present inside the Room	to : user_2 message : hello user_2	(private)user_1 : hello user_2	(private)user_1 : hello user_2	Pass
4	Pdf	share a pdf sent inside the cloud storage	The pdf must be present inside the storage	URL : "https//:project/testpdf.com"	display the pdf	pdf displayed inside the canvas	Pass

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

CONCLUSION

The rapid pace of global development has facilitated effortless connectivity, leading to the emergence of virtual conference rooms, as the next stage beyond traditional video calls. These rooms enable individuals to interact seamlessly as if they were physically present, fostering efficient sharing of work and collaboration. However, coordinating teams across different countries poses significant challenges, including limited progress visibility and difficulties in issue resolution. Moreover, remote work introduces a myriad of challenges such as increased isolation, home office expenses, overworking risks, productivity decline, distractions, workplace disconnection, and imbalanced work-life dynamics. Addressing these challenges is crucial for enhancing remote work efficiency and well-being.

FUTURE SCOPE

Looking ahead, the future scope of virtual conference platforms involves integrating meta humans an creating realistic office replicas foe virtual rental, allowing foe streamlined collaboration and efficient workspace design. This innovation holds the promise of revolutionizing markets by enhancing client-developer interaction and facilitating events such as product launches and fairs. By leveraging virtual environments, businesses can overcome geographical barriers and foster meaningful connections, ushering in a new era of remote work and collaborations.

APPENDICES

A.1 SDG GOALS

The implementation of GETO VR Workspace aligns with several United Nations Sustainable Development Goals (SDGs), contributing to towards a more sustainable and equitable future.

1. **SDG 4: Quality Education**

The system supports SDG 4 by enhancing the quality of the education and efficiency of the instructors' performance. By utilizing the VR meeting room, students could learn efficiently from the resource persons at ease and can also have practical experience.

2. **SDG 9: Industry, Innovation and Infrastructure**

GETO VR Workspace embodies SDG 9 by incorporating innovative technologies such as Virtual Reality, integration with cloud platforms and computer vision simulations such as hand tracking. It contributes to the sustainable and resilient infrastructure in the IT and education sector.

3. **SDG 12: Responsible consumption and production**

The system covers SDG 12 by monitoring the reducing the utilization of carbon footprints and the expenses spent to attend a physical meeting or a conference. In turn it reduces the consumption of fossil fuels such as petroleum and natural gas.

A.2 SOURCE CODE

PlayfabManager.cs

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using PlayFab;
using PlayFab.ClientModels;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
using TMPro;
public class PlayFabManager : MonoBehaviour
{
    string role;
    string username;
    [Header("Login")]
    public InputField l_username;
    public InputField l_password;
    [Header("SignUp")]
    public InputField s_email;
    public InputField s_password;
    public InputField s_username;
    public InputField s_confirmPass;
    public InputField s_phone;
    public TMP_Text greetingText;
    public TMP_Text pwd_err_text;
    public TMP_Text pwd_mismatch_text;
    [Header("Share_Data")]
```

```
    public MultiSceneDataCarrier multiSceneDataCarrier;
    private void Start()
    {
        role = "Employee";
        username = "Player";
        multiSceneDataCarrier.ResetValues();
        greetingText.enabled = false;
        pwd_err_text.enabled = false;
        pwd_mismatch_text.enabled = false;
    }
    //Login portion
    public void LoginButton()
    {
        var request = new LoginWithEmailAddressRequest {
            Email = l_username.text,
            Password = l_password.text,
            InfoRequestParameters = new GetPlayerCombinedInfoRequestParams
            {
                GetPlayerProfile = true
            }
        };
        PlayFabClientAPI.LoginWithEmailAddress(request, OnLoginSuccess, OnError);
    }
    void OnLoginSuccess(LoginResult result)
    {
        Debug.Log("Login Success");
        PlayFabClientAPI.GetUserData(new GetUserDataRequest(), OnDataReceived,
OnError);
        string u_name = null;
```

```
    if (result.InfoResultPayload.PlayerProfile != null)
    {
        u_name = result.InfoResultPayload.PlayerProfile.DisplayName;
        multiSceneDataCarrier.Username = u_name;
    }
    GetAccountInfo();
}
void OnDataReceived(GetUserDataResult result)
{
    if (result.Data != null && result.Data.ContainsKey("Role"))
    {
        multiSceneDataCarrier.Role = result.Data["Role"].Value;
    }
    if (result.Data != null && result.Data.ContainsKey("Email"))
    {
        multiSceneDataCarrier.Email = result.Data["Email"].Value;
    }
    if (result.Data != null && result.Data.ContainsKey("Phone"))
    {
        multiSceneDataCarrier.Phone = result.Data["Phone"].Value;
    }
}
public void RegisterButton()
{
    username = s_username.text;
    print(role);
    if(s_password.text == s_confirmPass.text)
    {
        var request = new RegisterPlayFabUserRequest
```

```
        {
            Email = s_email.text,
            Password = s_password.text,
            RequireBothUsernameAndEmail = false
        };
        PlayFabClientAPI.RegisterPlayFabUser(request, OnRegisterSuccess, OnError);
    }
    else
    {
        print("check your password");
        pwd_mismatch_text.enabled = true;
        pwd_err_text.enabled = false;
    }
}
void OnRegisterSuccess(RegisterPlayFabUserResult result)
{
    pwd_mismatch_text.enabled = false;
    pwd_err_text.enabled = false;
    Debug.Log("Registered successfully");
    StoreDisplayName();
    StoreUserData();
    greetingText.enabled = true;
}
public void StoreDisplayName()
{
    var request = new UpdateUserTitleDisplayNameRequest
    {
        DisplayName = username
    };
```

```
        PlayFabClientAPI.UpdateUserTitleDisplayName(request,
OnUpdateUserDisplayNameSuccess, OnUpdateUserDisplayNameFailure);
    }
    public void StoreUserData()
    {
        var role_Request = new UpdateUserDataRequest
        {
            Data = new Dictionary<string, string>
                {
                    {"Role", role},
                    {"Email", s_email.text},
                    {"Phone", s_phone.text}
                }
        };
        PlayFabClientAPI.UpdateUserData(role_Request, OnDataSent, OnError);
    }
    void OnDataSent(UpdateUserDataResult result)
    {
        Debug.Log("Data updated successfully");
    }
    public void ResetPasswordButton()
    {
        var request = new SendAccountRecoveryEmailRequest
        {
            Email = l_username.text,
            TitleId = "93755"
        };
        PlayFabClientAPI.SendAccountRecoveryEmail(request,
OnPasswordRecoverySuccess, OnError);
```

```
    }
    void OnPasswordRecoverySuccess(SendAccountRecoveryEmailResult result)
    {
        Debug.Log("Password reset mail sent successfully");
    }
    void OnError(PlayFabError error)
    {
        Debug.Log(error.GenerateErrorReport());
    }
    public void GetAccountInfo()
    {
        PlayFabClientAPI.GetAccountInfo(new GetAccountInfoRequest(),
OnGetAccountInfoSuccess, OnGetAccountInfoFailure);
    }
    public void OnGetAccountInfoSuccess(GetAccountInfoResult result)
    {
        string playFabId = result.AccountInfo.PlayFabId;
        Debug.Log("PlayFab ID: " + playFabId);
        multiSceneDataCarrier.PID = playFabId;
        SceneManager.LoadScene(1);
    }
    public void OnGetAccountInfoFailure(PlayFabError error)
    {
        Debug.LogError("Failed to retrieve account info: " + error.ErrorMessage);
    }
    private void OnUpdateUserDisplayNameSuccess(UpdateUserTitleDisplayNameResult
result)
    {
        Debug.Log("Display name updated successfully!");
```

```
        Debug.Log(username);
    }
    private void OnUpdateUserDisplayNameFailure(PlayFabError error)
    {
        Debug.LogError("Failed to update display name: " + error.ErrorMessage);
    }
    public void HandleRole(int _role)
    {
        if (_role == 0)
        {
            role = "Employee";
        }
        if (_role == 1)
        {
            role = "Manager";
        }
        if (_role == 2)
        {
            role = "Admin";
        }
    }
}
```

PlayerInfo.cs

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using PlayFab;
```

```
using PlayFab.ClientModels;
using UnityEngine.UI;
using TMPro;
public class Profile_Info : MonoBehaviour
{
    public MultiSceneDataCarrier multiSceneDataCarrier;
    public TMP_Text id;
    public TMP_Text Name;
    public TMP_Text email;
    public TMP_Text Gender;
    public TMP_Text phone;
    public TMP_Text profile_pic_name;
    public TMP_Text role;
    public string pid;
    public string username;
    public string userEmail;
    public string userGender;
    public string userPhone;
    public string userProfile;
    public string UserRole;
    void Start()
    {
        print("Username : " +  multiSceneDataCarrier.Username);
        print("Email : " + multiSceneDataCarrier.Email);
        pid = multiSceneDataCarrier.PID;
        username = multiSceneDataCarrier.Username;
        userEmail = multiSceneDataCarrier.Email;
        userGender = "Male";
        userPhone = multiSceneDataCarrier.Phone;
```

```
    userProfile = multiSceneDataCarrier.Username[0].ToString();
    UserRole = multiSceneDataCarrier.Role;
    updateProfile();
}
void updateProfile()
{
    if (pid.Length > 0)
    {
        id.text = pid;
    }
    if (username.Length > 0)
    {
        Name.text = username;
    }
    if (userEmail.Length > 0)
    {
        email.text = userEmail;
    }
    if (userGender.Length > 0)
    {
        Gender.text = userGender;
    }
    if (userPhone.Length > 0)
    {
        phone.text = userPhone;
    }
    if (userProfile.Length > 0)
    {
        profile_pic_name.text = userProfile;
```

```
        }
        if (UserRole.Length > 0)
        {
            role.text = UserRole;
        }
    }
}
```

MultiSceneDataCarrier.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
[CreateAssetMenu]
public class MultiSceneDataCarrier : ScriptableObject
{
    [SerializeField]
    private string userName;
    private string player_id;
    [SerializeField]
    private string email;
    [SerializeField]
    private string role;
    [SerializeField]
    private string phone;
    public string Username
    {
        get { return userName; }
        set { userName = value; }
    }
    public string PID
```

```
    {
        get { return player_id; }
        set { player_id = value; }
    }
    public string Email
    {
        get { return email; }
        set { email = value; }
    }
    public string Role
    {
        get { return role; }
        set { role = value; }
    }
    public string Phone
    {
        get { return phone; }
        set { phone = value; }
    }
    public void ResetValues()
    {
        userName = "";
        player_id = "";
        email = "";
        role = "";
        phone = "";
    }
}
```

Webcamera.cs

```
using System.Collections;
using
System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class webCamera :
MonoBehaviour
{
   public RawImage img =
default;
   private WebCamTexture
webCam;
   void Start()
   {
      webCam = new WebCamTexture();
      if (!webCam.isPlaying) webCam.Play();
      img.texture = webCam;
   }
}
```

PhotonChatManager.cs

```
using System.Collections;
using Photon.Chat;
using System.Collections.Generic;
using UnityEngine;
using ExitGames.Client.Photon;
using Photon.Pun;
```

```
using Photon.Realtime;
using UnityEngine.UI;
using TMPro;
public class PhotonChatManager : MonoBehaviour,IChatClientListener
{
    #region Setup
    public CreateorjoinRoom createorjoinRoom;
    [SerializeField] GameObject joinChatButton;
    ChatClient chatClient;
    bool isConnected;
    [SerializeField] string username;
    public void UsernameOnValueChange(string valueIn)
    {
        username = valueIn;
    }
    public void ChatConnectOnClick()
    {
        isConnected = true;
        chatClient = new ChatClient(this);
chatClient.Connect(PhotonNetwork.PhotonServerSettings.AppSettings.AppIdChat,
PhotonNetwork.AppVersion, new Photon.Chat.AuthenticationValues(username));
        Debug.Log("Connenting");
    }
    #endregion Setup
    #region General
    [SerializeField] GameObject chatPanel;
    string privateReceiver = "";
    string currentChat;
    [SerializeField] TMP_InputField chatField;
```

```
[SerializeField] TMP_Text chatDisplay;
// Start is called before the first frame update
void Start()
{
}
// Update is called once per frame
void Update()
{
    if (isConnected)
    {
        chatClient.Service();
    }
    if (chatField.text != "" && Input.GetKey(KeyCode.Return))
    {
        SubmitPublicChatOnClick();
        SubmitPrivateChatOnClick();
    }
}
#endregion General
#region PublicChat
public void SubmitPublicChatOnClick()
{
    if (privateReceiver == "")
    {
        chatClient.PublishMessage("RegionChannel", currentChat);
        chatField.text = "";
        currentChat = "";
    }
```

```
}
public void TypeChatOnValueChange(string valueIn)
{
    currentChat = valueIn;
}
#endregion PublicChat
#region PrivateChat
public void ReceiverOnValueChange(string valueIn)
{
    privateReceiver = valueIn;
}
public void SubmitPrivateChatOnClick()
{
    if (privateReceiver != "")
    {
        chatClient.SendPrivateMessage(privateReceiver, currentChat);
        chatField.text = "";
        currentChat = "";
    }
}
#endregion PrivateChat
#region Callbacks
public void DebugReturn(DebugLevel level, string message)
{
    //throw new System.NotImplementedException();
}
public void OnChatStateChange(ChatState state)
{
    if (state == ChatState.Uninitialized)
```

```
        {
            isConnected = false;
            joinChatButton.SetActive(true);
            chatPanel.SetActive(false);
        }
    }
    public void OnConnected()
    {
        Debug.Log("Connected");
        joinChatButton.SetActive(false);
        chatClient.Subscribe(new string[] { "RegionChannel" });
        foreach (Player player in PhotonNetwork.PlayerList)
        {
            print(player.NickName);
        }
    }
    public void OnDisconnected()
    {
        isConnected = false;
        joinChatButton.SetActive(true);
        chatPanel.SetActive(false);
    }
    public void OnGetMessages(string channelName, string[] senders, object[]
messages)
    {
        string msgs = "";
        for (int i = 0; i < senders.Length; i++)
        {
            msgs = string.Format("{0}: {1}", senders[i], messages[i]);
```

```
            chatDisplay.text += "\n" + msgs;
            Debug.Log(msgs);
        }
    }
    public void OnPrivateMessage(string sender, object message, string channelName)
    {
        string msgs = "";
        msgs = string.Format("(Private) {0}: {1}", sender, message);
        chatDisplay.text += "\n " + msgs;
        Debug.Log(msgs);
    }
    public void OnStatusUpdate(string user, int status, bool gotMessage, object
message)
    {
        throw new System.NotImplementedException();
    }
    public void OnSubscribed(string[] channels, bool[] results)
    {
        chatPanel.SetActive(true);
    }
    public void OnUnsubscribed(string[] channels)
    {
        throw new System.NotImplementedException();
    }
    public void OnUserSubscribed(string channel, string user)
    {
        throw new System.NotImplementedException();
    }
    public void OnUserUnsubscribed(string channel, string user)
```

```
    {
        throw new System.NotImplementedException();
    }
    #endregion Callbacks
}
```

CreateorjoinRoom.cs

```
using Photon.Pun;
using Photon.Realtime;
using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;
public class CreateorjoinRoom : MonoBehaviourPunCallbacks
{
    public TMP_InputField createInput;
    public TMP_InputField joinInput;
    public TMP_InputField Uname;
    //public GameObject player;
    private void Start()
    {
        PhotonNetwork.ConnectUsingSettings();
    }
    public override void OnConnectedToMaster()
    {
        print("Connected to server");
        PhotonNetwork.JoinLobby();
    }
    public override void OnJoinedLobby()
```

```
    {
        print("Joined the lobby");
    }
    public void CreateRoom()
    {
        PhotonNetwork.CreateRoom(createInput.text);
    }
    public void JoinRoom()
    {
        PhotonNetwork.JoinRoom(joinInput.text);
    }
    public override void OnJoinedRoom()
    {
        PhotonNetwork.NickName = Uname.text;
        PlayerPrefs.SetString("UserName", Uname.text);
        PhotonNetwork.LoadLevel("SampleScene");
        Debug.Log(PhotonNetwork.PlayerList.Length.ToString());
    }
```

ShowKeyboard.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using Microsoft.MixedReality.Toolkit.Experimental.UI;
public class ShowKeyboard : MonoBehaviour
{
    private TMP_InputField message;
    public float distance = 0.5f;
```

```
    public float verticalOffset = -0.5f;
    public Transform positionSource;
    public GameObject keyboard;
    void Start()
    {
        keyboard.SetActive(false);
        message= GetComponent<TMP_InputField>();
        message.onSelect.AddListener(x => openKeyboard());
    }
    public void openKeyboard()
    {
        keyboard.SetActive(true);
        NonNativeKeyboard.Instance.InputField= message;
        NonNativeKeyboard.Instance.PresentKeyboard(message.text);
        //direction of the cam
        Vector3 direction = positionSource.forward;
        direction.y = 0;
        direction.Normalize();

        Vector3 targetPosition= positionSource.position + direction * distance+
Vector3.up * verticalOffset;
        NonNativeKeyboard.Instance.RepositionKeyboard(targetPosition);
    }
}
```

VideoDisplay.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

```
using UnityEngine.UI;
using TMPro;
using System.IO;
using Unity.Mathematics;
public class videodisplay : MonoBehaviour
{
    public GameObject v;
    public string vurl;
    public GameObject qvideo;
    public GameObject pan;
    void Start()
    {
        Button b=v.GetComponent<Button>();
        vurl=videolist.URLs[0];
        b.GetComponentInChildren<TMP_Text>().text=Path.GetFileName(vurl);
        b.onClick.AddListener(() => vopen(vurl));
        displaylist();
    }
    void displaylist()
    {
        for (int i=1;i<videolist.URLs.Length;i++)
        {
            if (videolist.URLs[i] != null)
            {
                GameObject newbutton = Instantiate(v, new
Vector3(v.transform.position.x, v.transform.position.y - i * 40,
v.transform.position.z), quaternion.identity);
                newbutton.transform.SetParent(pan.transform);
                Button b1 = newbutton.GetComponent<Button>();
```

```
            string vu = videolist.URLs[i];
            b1.GetComponentInChildren<TMP_Text>().text = Path.GetFileName(vu);
            b1.onClick.AddListener(() => vopen(vu));
        }
    }
}
public void vopen(string uv)
{
    ShareVi sharevideoscript=qvideo.GetComponent<ShareVi>();
    sharevideoscript.ShareVideoURL(uv);
}
}
```

A.3 SCREENSHOTS

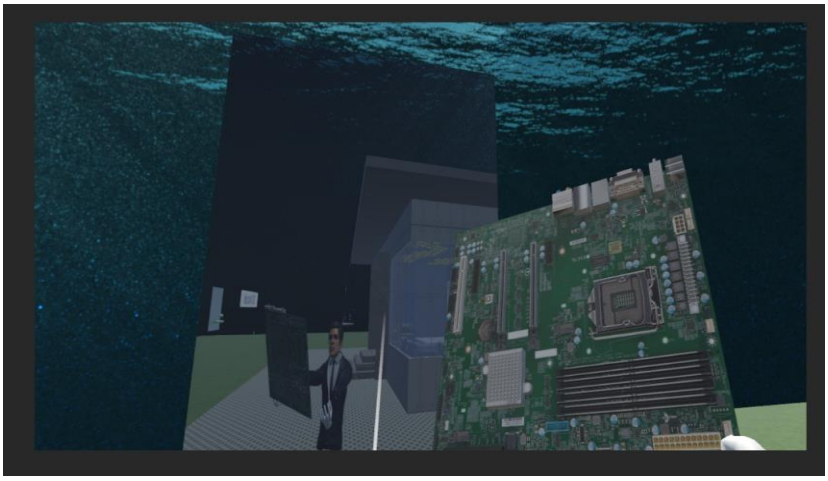


Figure 7.1- 3D object

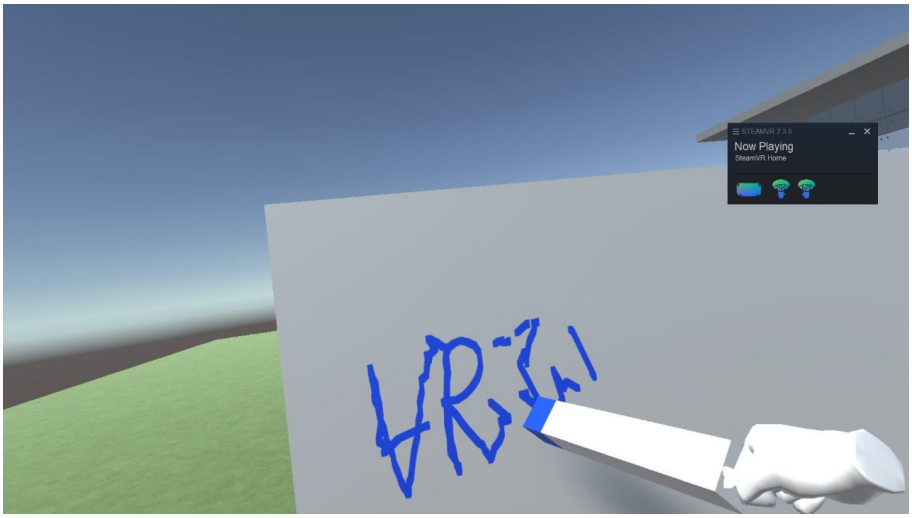


Figure 7.2- White board

Figure 7.3- Ray Tracing

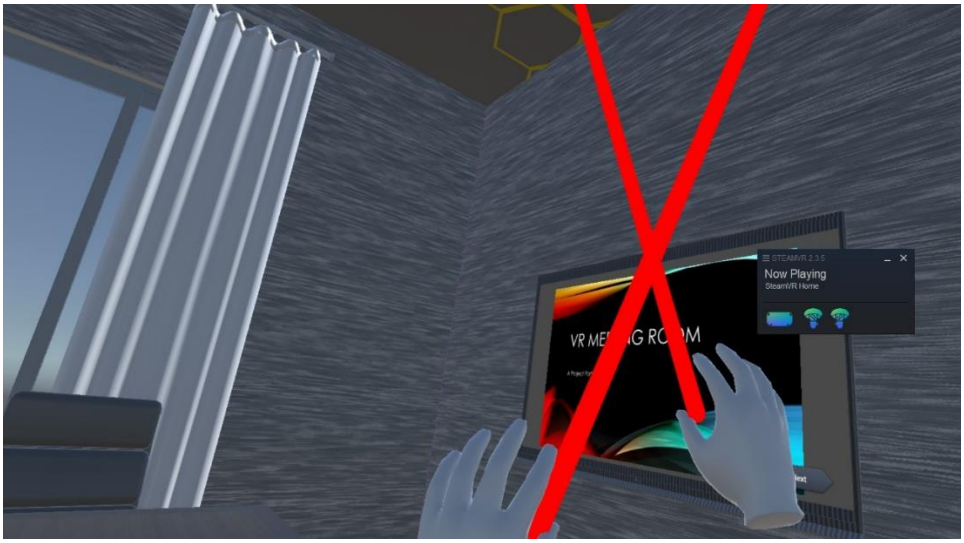
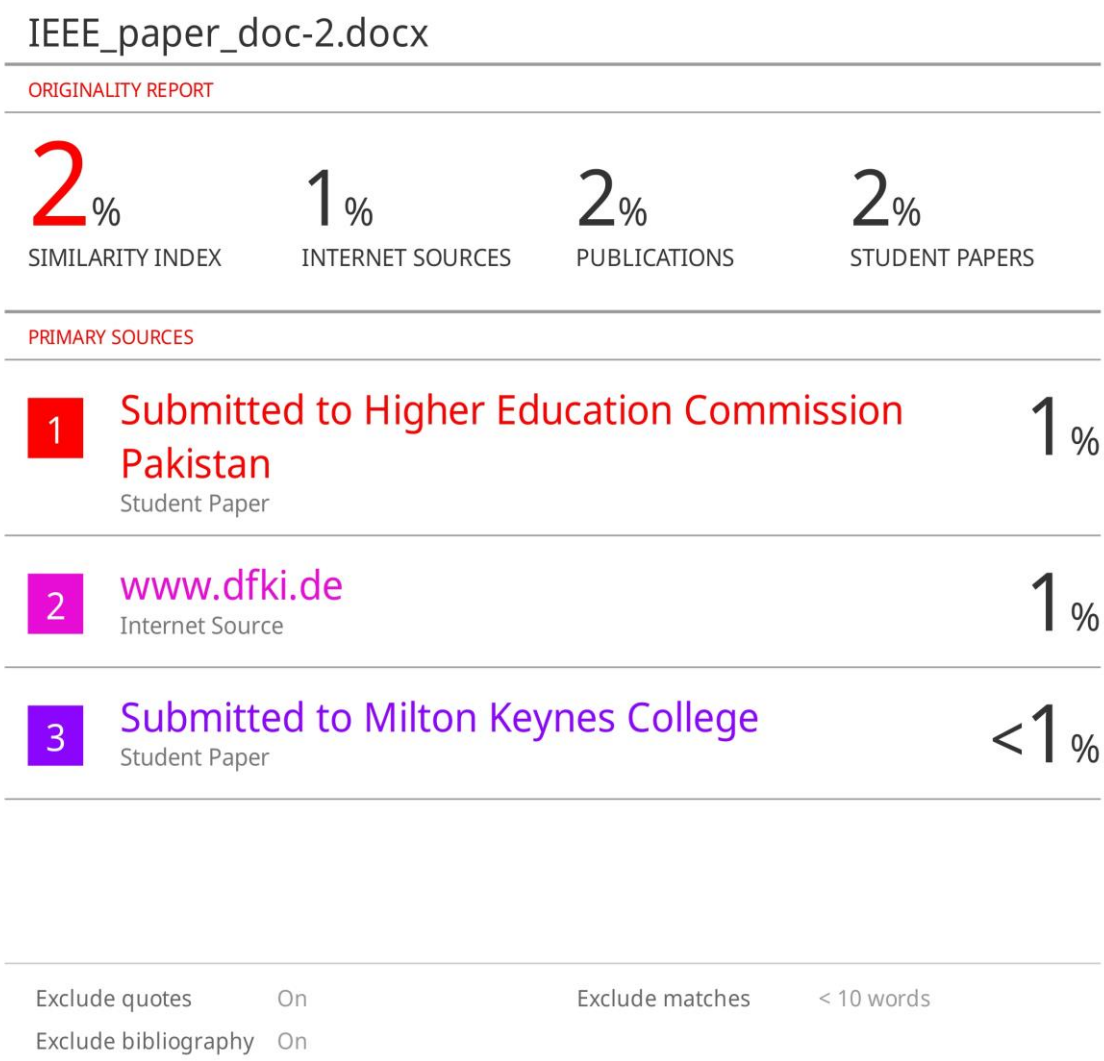


Figure 7.4- PDF importer

A.4 PLAGIARISM REPORT



IEEE_paper_doc-2.docx

ORIGINALITY REPORT

2%	1%	2%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	1%
2	www.dfki.de Internet Source	1%
3	Submitted to Milton Keynes College Student Paper	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

A.5 PAPER PUBLICATION

REFERENCES:

1. Osvaldo Gervasi, Damiano Perri, Marco Simonetti, "Empowering Knowledge With Virtual and Augmented Reality", IEEE Access (Volume 11), 12 Dec. 2023.
2. Osvaldo Gervasi, Damiano Perri, Marco Simonetti, "The Metaverse: Applications, Concerns, Technical Challenges, Future Directions and Recommendations", IEEE Access (Volume 11), 04 Oct. 2023.
3. Abbas M. Al-Ghaili, Hairoladenan Kasim, Naif Mohammed Al-Hada, Zainuddin Bin Hassan, Marini Othman, Jakir Hussain Tharik, Rafiziana Md. Kasmani, Ibraheem Shayea, "A Review of Metaverse's Definitions, Architecture, Applications, Challenges, Issues, Solutions, and Future Trends", IEEE Access (Volume 10), 30 Nov. 2022.
4. Ashwaq Z. Amat, Deeskha Adiani, Mahrukh Tauseef, Michael Breen, Spencer Hunt, Amy R Swanson, Amy S. Weitlauf, Zachary E. Warren, Nilanjan Sarkar, "Design of a Desktop Virtual Reality-Based Collaborative Activities Simulator (ViRCAS) to Support Teamwork in Workplace Settings for Autistic Adults", IEEE Transactions on Neural Systems and Rehabilitation Engineering (Volume 31), 27 Apr. 2023.
5. Catarina Moreira, Francisco P. M. Simoes, Mark J. W. Lee, Ezqruiel R. Zorzal, Robert W. Lindeman, Joao Madeiras Pereira, Kyle Johnsen, Joaquim Jorge "Toward VR in VR: Assessing Engagement and Social Interaction in a Virtual Conference", IEEE (Volume 11), 29 Dec. 2022.
6. Ariana Remmel "Scientists want virtual meetings to stay after the COVID pandemic" Nature, 02 Mar. 2021.
7. Alexander Schafer, Gerd Reis, Didier Stricker "Towards Collaborative Photorealistic VR Meeting Rooms" Mensch und Computer 2021.
8. Ramon Olivia, Alejandro Beacco, Jaime Gallego, Raul Gallego Abellan, Mei Siater, "The Making of a Newspaper Interview in Virtual Reality: Realistic Avatars, Philosophy, and Sushi" IEEE Computer Graphics and Applications (Volume 43), Nov-Dec. 2023.

9. Alannah Lax-Vanek, Claire Davies “A Step in the Right Direction: Evaluating the Effectiveness of Customized Stepping Game Software and Balance Boards for Balance Rehabilitation Therapy and Measurement” 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), July 2023.
10. Chaiyong Ragkhitwetsagul, Morakot Choetkiertikul, Apirak Hoonlor, Mores Prachyabrued “Virtual Reality for Software Engineering Presentations” 29th Asia-Pacific Software Engineering Conference (APSEC), Dec. 2022.
11. Martin Stancek[a], Ivan Polasek[b,a,*], Tibor Zalabai[a] , Juraj Vincur[b], Rodi Jolak[c], Michel Chaudron[d] “Collaborative Software Design and Modeling in Virtual Reality” 29th Nov. 2023.