The Use of Extended Reality and Machine Learning to Improve Healthcare and Promote GreenHealth

André Silva Moreira

This Thesis was presented to the School of Technology and Management of the Polytechnic University of Bragança to obtain the Master Degree in Informatics

.

Advisors:

Doctor Carlos Filipe Campos Rompante da Cunha

Doctor Rui Pedro Sanches de Castro Lopes

Bragança

December 2023

The Use of Extended Reality and Machine Learning to Improve Healthcare and Promote GreenHealth

André Silva Moreira

This Thesis was presented to the School of Technology and Management of the Polytechnic University of Bragança to obtain the Master Degree in Informatics

.

Advisors:

Doctor Carlos Filipe Campos Rompante da Cunha

Doctor Rui Pedro Sanches de Castro Lopes

Bragança

December 2023

We learn from failure, not from success!

Bram Stoker, *Dracula*

If you only do what you can do, you'll never be more than you are now.

Master Shifu, *Kung Fu Panda*

The School of Technology and Management of Bragança is not responsible for the opinions expressed in this thesis.

This work was funded by the European Regional Development Fund (ERDF) through the Regional Operational Program North 2020, within the scope of Project GreenHealth: Digital strategies in biological assets to improve wellbeing and promote green health, Norte-01-0145--FEDER-000042.

Dedication

I dedicate this work to my grandmother on mother side, who took care of me since childhood, through hard times, sickness and disagreements, who in the last years started to suffer from amnesia and demency, who now needs to stay in a nursing home.
While not being able to understand this, thank you.

Acknowledgement

I want to thank my parents for their support, in all forms, and to my grandfather who also supported me and always found time to give his "old timer" motivation quotes since the Bachelor till now, final moments of my Master. I also want to thank my supervisors, Professor Dr. Rui Pedro Lopes for his guidance, and Professor Dr. Carlos Rompante Cunha, who motivated me to adventure into a Master Degree and also made this work possible through his guidance, support and patience. I thank the Applied Management Research Unit (Unidade de Investigação Aplicada em Gestão (UNIAG)) for accepting me as Research Fellow in the GreenHealth Project and for providing the tools necessary which made the foundations for this work, as follow I thank the UNIAG personal for their warm welcome. Finally I want to thank the professors and staff at the School of Communication, Public Management and Tourism of Mirandela (Escola Superior de Comunicação, Administração e Turismo (EsACT)) for the nice and friendly environment that I found every day during my work.

Resumo

Com a Quarta Revolução Industrial, a propagação da Internet das Coisas, o avanço nas áreas de Inteligência Artificial e de Machine Learning até à migração para a Computação em Nuvem, o termo "Ambientes Inteligentes" cada vez mais deixa de ser uma idealização para se tornar realidade. Da mesma forma as tecnologias de Realidade Extendida também elas têm aumentado a sua presença no mundo tecnológico após um "período de hibernação", desde a popularização do conceito de Metaverse assim como a entrada das grandes empresas informáticas como a Apple e a Google num mercado onde a Realidade Virtual, Realidade Aumentada e Realidade Mista eram dominadas por empresas com menos experiência no desenvolvimento de sistemas (e.g. Meta), reconhecimento a nível mundial (e.g. HTC Vive), ou suporte financeiro e confiança do mercado. Esta tese tem como foco o estudo do potencial uso das tecnologias de Realidade Estendida de forma a promover Saúde Verde assim como seu uso em Hospitais Inteligentes, uma das variantes de Ambientes Inteligentes, incorporando Machine Learning e Computer Vision, como ferramenta de suporte e de melhoria de cuidados de saúde, tanto do ponto de vista do profissional de saúde como do paciente, através duma revisão literarária e análise da atualidade. Resultando na elaboração de um modelo conceptual com a sugestão de tecnologias a poderem ser usadas para alcançar esse cenário selecionadas pelo seu potencial, sendo posteriormente descrito o desenvolvimento de protótipos de partes do modelo conceptual para Óculos de Realidade Extendida como validação de conceito.

Palavras-chave: Realidade Estendida, Cuidados de Saúde, Hospital Inteligente, Machine Learning, Computer Vision, Internet das Coisas, Saúde Verde

Abstract

With the Fourth Industrial Revolution, the spread of the Internet of Things, the advance in the areas of Artificial Intelligence and Machine Learning until the migration to Cloud Computing, the term "Intelligent Environments" increasingly ceases to be an idealization to become reality. Likewise, Extended Reality technologies have also increased their presence in the technological world after a "hibernation period", since the popularization of the Metaverse concept, as well as the entry of large computer companies such as Apple and Google into a market where Virtual Reality, Augmented Reality and Mixed Reality were dominated by companies with less experience in system development (e.g. Meta), worldwide recognition (e.g. HTC Vive) or financial support and trust in the market. This thesis focuses on the study of the potential use of Extended Reality technologies in order to promote GreenHealth as well as their use in Smart Hospitals, one of the variants of Smart Environments, incorporating Machine Learning and Computer Vision, as a tool to support and improve healthcare, both from the point of view of the health professional and the patient, through a literature review and analysis of the current situation. Resulting in the elaboration of a conceptual model with the suggestion of technologies that can be used to achieve this scenario selected for their potential, and then the development of prototypes of parts of the conceptual model for Extended Reality Headsets as concept validation.

Palavras-chave: Extended Reality, Healthcare, Smart Hospital, Machine Learning, Computer Vision, Internet of Things, GreenHealth

Contents

Acronyms

1D One-Dimensional. 19

2D Two-Dimensional. 19, 57, 71, 92, 106

3D Three-Dimensional. 12, 26, 27, 57, 59, 89–92, 96, 106, 110, 111

3DoF Three Degrees of Freedom. 6

6DoF Six Degrees of Freedom. 6

AI Artificial Inteligence. 21, 25, 30, 47

AIDC Automatic Identification and Data Capture. 19

ANSI American National Standards Institute. 18

API Application Programming Interface. 19, 43, 46, 51–53, 56, 61, 65, 73–76, 78, 82, 83, 89, 98, 99, 105, 106, *Glossary:* API

AR Augmented Reality. 5, 7–10, 13, 19, 21, 23, 25, 27, 36, 37, 39, 40, 57–59, 66, 111

ASGI Asynchronous Server Gateway Interface. 75

AWS Amazon Web Services. 41, 73, 74

CAGR Compound Annual Growth Rate. 15

CAVE Cave Automatic Virtual Environment. 7, 28

CCD Continuity of Care Document. 18

CDA Clinical Document Architecture. 18

CDMS Clinical Document Management Systems. 16

CNN Convolutional Neural Network. 81, 82

CPU Central Processing Unit. 40, 68, 74

CRUD Create, Read, Update, Delete. 49, 51, 75, 110

CSR Client-Side Rendering. 60

CTB Cognitive Behavioral Therapy. 30

CV Computer Vision. 8, 19–21, 23, 25, 33, 60, 61, 99, 109

DB Database. 23, 29, 30, 34, 48–52, 56, 69–73, 75, 78–80, 84, 87, 94, 97, 110, E1

DBaaS Database-as-a-Service. 69

DS Data Science. 17, 23, 47, 49, 69, 70, 110

EBS Enterprise Service Bus. 77

EHR Electronic Health Record. 2, 15–19, 33

EMR Electronic Medical Record. 16, 17

EsACT Escola Superior de Comunicação, Administração e Turismo. xiii, 39

FHIR Fast Healthcare Interoperability Resources. 19, 33, 110

FoV Field of View. 40

FPS Frame Per Second. 87

FTP File Transfer Protocol. 43

GDS Graph Data Science. 69, 72

GPS Global Positioning System. 8, 28, 35

GPU Graphics Processing Unit. 68

gRPC Google Remote Procedure Call. 43, 52, 57, 78–81, 84, 100

GUI Graphic User Interface. 45, 71, 72

HIS Hospital Information Systems. 16

HL2 HoloLens 2. 5, 9, 13, 25, 26, 37, 39, 40, 57, 58, 65, 66, 68, 73, 80, 94–96, 98, 100–102

HL7 Health Level 7. 3, 18, 19

HMD Head-mounted display. 6, 7, 9, 10, 12, 25, 26, 28, 29, 31, 35–37, 39–41, 51, 58, 64, 67, 68, 83, 89, 90, 94, 98, 102, 105

HTTP Hypertext Transfer Protocol. 43, 52, 53, 98, 99

IaaS Infrastructure as a Service. 42

ICT Information and Communication Technologies. 16

IDE Integrated Development Environment. 10, 42, 98, 102

IL2CPP Intermediate Language To C++. 66

IoMT Internet of Medical Things. 14

IoT Internet of Things. 1, 13, 14, 33, 34, 50, 54–56, 72, 73, 97

IP Internet Protocol. 55, 61

IT Information Technology. 18, 41

JIS Japanese Industrial Standards. 20

JSON JavaScript Object Notation. 49, 51, 56, 57, 84, 86, 87, 91, 98

JVM Java Virtual Machine. 42

JWT JSON Web Token. 77, 83, 105, *Glossary:* JWT

LPB Local Binary Pattern. 80, 81, 101

ML Machine Learning. 2, 8, 17, 21, 23, 25, 30, 33, 47, 64, 68, 74, 75, 79, 81, 109

ML-Agents Machine Learning Agents Toolkit. 69

MQTT Message Queuing Telemetry Transport. 55

MR Mixed Reality. 3, 5, 7–10, 13, 19, 21, 23, 25, 26, 35, 37, 39, 40, 45, 57, 63, 64, 69, 80, 87, 94, 95, 97, 98, 105, 109, 111, E4

MRI Magnetic resonance imaging. 16

MRTK Mixed Reality Toolkit. 57, 58, 95, 97, 111

MS Microservice. 45, 47, 52, 61, 74, 77–84, 96, 98–100, 109

NHS National Health Service. 16

NoSQL Not Only SQL. 48–50, 71, 87

OCD Obsessive-Compulsive Disorder. 30

OEM Original Equipment Manufacturer. 67

ONNX Open Neural Network Exchange. 68, 69

OOP Object-Oriented Programming. 46, 47, 50

OSI Open Systems Interconnection. 18

OST Optical See-Through. 9

PaaS Software as a Service. 42

PACS Picture Archiving And Communication Systems. 16

PDF Portable Document Format. 48

PHR Personal Health Record. 16, 17

Protobuf Protocol Buffers. 52, 56, 57, 79, 85, 91, 100

Pub/Sub Publish/Subscribe. 53, 54, 72, 84–86

QR Quick Response. 8, 19, 20, 27, 35–37, 47, 50, 62, 64, 72, 79, 80, 83, 84, 95, 96, E4

RAM Random Access Memory. 51, 73, 74

REST Representational State Transfer. 51, 52, 56

RPC Remote Procedure Call. 51, 52

SaaS Platform as a Service. 42

SDK Software Development Kit. 10, 26, 27, 58, 59, 67, 93, 94, E3, *Glossary:* SDK

SEO Search Engine Optimization. 60

SH Smart Hospital. 33

SLAM Simultaneous Localization and Mapping. 8, 28

SOA Service-Oriented Architecture. 77

SOAP Simple Object Access Protocol. 51

SQL Structured Query Language. 48–50, 70, 87

SSE Server-Sent Events. 53, 86, 107

SSG Static Site Generation. 60

SSH Secure Shell. 43, 61

SSID Service Set Identifier. 27

SSR Server-Side Rendering. 60

TCP Transmission Control Protocol. 52, 78

UI User Interface. 3, 45, 57, 60, 94, 97, 104, 110

UNIAG Unidade de Investigação Aplicada em Gestão. xiii, 39

UPC Universal Product Code. 19

URL Uniform Resource Locator. 82, 86

UUID Universally Unique Identifier. 71, 72, 83, 84, 86, 93, 105

UWP Universal Windows Platform. 42, 65, 80, 94, 111

UX User Experience. 3, 42, 57, 104, 110

VM Virtual Machine. 42, 69, 73, 74

VPN Virtual Private Network. 61, 78

VR Virtual Reality. 5–10, 12, 21, 23, 27–30, 36, 37, 39–41, 50, 58, 63, 64, 67, 68, 71, 72, 84–87, 89–91, 105, 106, 109–111, E2

VRET Virtual Reality Exposure Therapy. 30, 31

WinRT Windows Runtime. 65, 66, 68, 80, 95, 100, 102

WS WebSockets. 43, 47, 51–53, 61, 72, 74, 75, 82–86, 89–91, 93, 94, 96, 99, 105, 107, 111

WVSMS Wireless Vital Signs Monitoring System. 15

WWW World Wide Web. 61

XML Extensible Markup Language. 56, 57

XR Extended Reality. 1, 2, 5, 6, 8–13, 20, 21, 23, 27, 29, 31, 33, 35, 36, 39, 40, 43, 44, 46, 51, 57–60, 63, 68, 83, 86

YOLO You Only Look Once. 79, 82

List of Figures

List of Tables

List of Listings

Chapter 1 Introduction

As time goes by, healthcare institutions become increasingly overloaded and understaffed, and current professionals grow more tired and find it challenging to maintain good performance and motivation, not only in their work but also in their interaction with patients, which has become more notable during the recent pandemic of Covid-19 [1], at the same time the patients also become more tired of the time consumed and needed to take care of their physical health at cost of travels, time and their mental state.

Although the new technologies have being used to support and improve healthcare, since the invention of the computer which set free the medical staff from carrying loads of papers and processing information by hand, the advanced medical machines like medical robots that support in hard and complex tasks such as operations till the more recent Internet of Things (IoT) devices that are starting to be more present in those institutions, the Extended Reality (XR) technologies are still hardly being seen in use aside from research projects or for specific and limited use cases such as exercises and medical training.

XR has the potential to improve well-being and GreenHealth of both patients and professionals by enabling scenarios and actions that others technologies were not able to while requiring a low cost, both to the users as well to the environment.

In overall, XR has shown a diversification of use cases such as (real-time) data visualization, remote collaboration, safer and less costly simulations, training and education, even more advanced ones such as systems remote control [2], use cases which can be also applied in the Healthcare sector with the potential to solve many of the current identified problems in the same way that the previous inventions, like the computers, also solved older problems.

Among those problems is the current inability of the institutions to answer the growth of life

expectancy and the increase of elders with the need of continue care which normally is solved by gathering them in nursing homes, however each patient has different health conditions and needs, but also different habits and ways of life, not all want to stay seated or enclosed in a room, this however increases the work and mental burden of the caregivers.

To better understand the current situation, in this thesis is done a review of the State of the Art related to the use of the XR technology in the Healthcare sector, having special attention to those which support caregivers work and focused on elders. In parallel with the State of the Art is also made an analyse of what is considered the future of Healthcare and what terms and technologies may be allied with it.

After understanding the current results, is analysed what can be improved and how, through the development of a conceptual model with base on the combination of XR technologies and Machine Learning (ML) to improve Healthcare, being identified and described potential technologies to be used and implemented prototypes as proof of concept.

1.1 Overall Goal

The goal of this thesis is to study the use of XR applications with ML to improve Healthcare, from a technological point-of-view, and well-being while promoting GreenHealth, among other concepts and technologies that are considered part of the future of Healthcare Institutions.

The scenario envisaged is one that depends on interoperability between healthcare institutions, as is to be expected in an Electronic Health Record (EHR) structure. To achieve this, an analysis of the most recent technologies is required, as well as the implementation and experimentation to validate the concepts. Additionally, specific categories of end-users have been assigned to each type of XR application, considering factors such as portability, affordability, potential use, and constraints:

- **Mixed Reality Application:** Focused on being used in the Institutions by Health Professionals, such as Medics, Nurses or Caregivers, should act as a intangible, free-hands "Desktop Application", easy access to EHR and general data visualization. Should also take advantage of the device capacities such as Camera Capture, connection to the Network and

"Multiplayer" experiences, Mixed Reality (MR) devices such as HoloLens 2, Magic Leap 2 or even the Apple Vision Pro can be considered too expensive for the common user;

- **Virtual Reality Application:** Used on patients to conduct exams, virtual consultations, or treatments. For this thesis, special attention was given to the use of eye tracking with the HTC Vive Pro Eye, which utilizes Tobii XR technology;

- **Augmented Reality Application:** For occasional and general use but with a focus on the public, like Patients family and hospital visitors;

To support these applications, a back-end was also developed, along with additional features on a web application to complement them.

It is important to mention that the focus of this thesis and the developed prototypes involves the exploration of technologies and the identification of potential uses, as well as experimentation with different approaches, being most of the used technologies unknown to the author till the date, which required a learning process that consisted on learning base on the necessity.

Aspects such as User Interface (UI), User Experience (UX), the use of standards like Health Level 7 (HL7) (2.6.2), or the development of a ready-to-use application was not defined as goals or criteria, being that part of a future work where it was already laid the foundations.

1.2 Thesis Structure

This Thesis is divided in 7 chapters, including the Introduction. The Chapter 2 and 3 are theoretic driven, in the Chapter 2 is introduced some concepts and terms to the reader which are necessary to a better understanding of the project, may the reader be unfamiliar with certain fields, while in the Chapter 3 is made a scientific and commercial review about the use of Extended Reality in the Healthcare sector.

In the Chapter 4 is presented a novel conceptual model which combines topics from the Chapter 2 and have interest to work as a bridge between products of the Chapter 3. In the Chapter 5 and 6 is approached more practical topics, the Chapter 5 presents the development environment and the technologies chosen after research and tests, while in the Chapter 6 is explained the prototypes

and components developed of the conceptual model presented in the Chapter 4, while using the technologies from the Chapter 5. The Chapter 7 presents an overview of the project results, a self-evaluation of the process and remarks about future work.

Chapter 2 Background

In this chapter will be presented and explained necessary concepts, some of them are directly present on the project such as *Extended Reality* (2.1) while others are essential, like *The Metaverse* (2.2), for a better understanding of the project. The present explanations are a mixed of general knowledge and interpretation of the author.

2.1 Extended Reality

XR is a term used to group the technologies Virtual Reality (VR), Augmented Reality (AR), MR (Figure 2.1) and all future immersive technologies which merge the physical and virtual worlds, is used to promote the new realities [3], a generic expression used as umbrella [4], while also being used to describe devices like the VIVE XR Elite or Meta Quest Pro that are able to provide a mix of experiences instead of being limited to only one reality like the HoloLens 2 (HL2) or the old Oculus Quest 1.

However when researched in depth, the XR technologies can become a more complex and intrigued field that it is seen on the surface, starting from the term "Extended Reality" itself till categories and sub-categories [5], in this thesis will be only considered the three more common.

- **Virtual Reality:** VR is the oldest and most well established term among the XR technologies, the concept of VR can be mapped till the years 1980s being first described by Sir Charles Wheatstone, yet only started to become what is seen nowadays in 2012 with the kickstart of Oculus for the "first truly immersive virtual reality headset" [6]. VR can be

found being used on many fields like the entertainment industry, education, medicine or job's training [7]. In simple terms, VR consist in being surrounded by the digital world, where the humans are guests and able to interact with [5].

Figure 2.1: Augmented Reality, Mixed Reality and Virtual Reality (Adapted from [8][9][10])

This experience is enabled through the use of Head-mounted display (HMD), which can be standalone like the Meta Quest 2 or wired like the HTC Vive Pro, being used controllers to enable interaction, however some more recent advances like the technology "Passthrough" enables the use of hand tracking instead. Such immersion its also possible thanks to head and body position tracking [11] as well by having Six Degrees of Freedom (6DoF), which refers to the tracking of both position and rotation of the user (Figure 2.2), instead of Three Degrees of Freedom (3DoF) which only tracks rotation (e.g. Google Cardboard), this can involve the use of sensors, like the base stations of HTC Vive Pro, which makes necessary the allocation of a space, zone to put the sensors and, optionally, the definition of a room-scale play area, being this the term used by Vive.

VR, as concept instead of XR technology, can also be divided in three categories depending on the level of immersion enabled:

- **Non-Immersive:** Use of flat screen while simulating the reality on a digital world (e.g. Life Simulator);
- **Semi-immersive:** Can use several large screens around the user, normally together

with controllers alike the "real-world controller" (e.g. Fly Simulator);

– **Fully-immersive:** Makes use of HMD with controllers/hand-tracking or use of Cave Automatic Virtual Environment (CAVE) systems;

When it comes to immersion in VR, certain aspects must be taken in account such as sense of presence, movement and how the user control/interacts with the digital while avoiding cybersickness like Vection Illusion. Spatial audio, immersive diegetic/non-diegetic sounds and, when possible, olfactory sensations or even the use of haptic feedback are also important factors to create a good VR experience and to immerse the user into the digital environment. As well the use of similar scales to the real world, like for example, one unit, the measure used in Unity, equals to one meter in real-life.

Figure 2.2: Degrees of Freedom in Virtual Reality [12]

- **Augmented Reality:** Like VR, AR is also already established on the market and due to rather easier accessibility for not requiring a specific kind of device like VR or MR it can

be seen being used in domains where VR has yet to establish like manufacturing [13], being also a preferential choice of industries when looking for entering in the XR world. Most of the recent smartphones and tablets are able to run AR applications and the capacity of supporting this technology is starting to be considered a "must have" by the manufacturers.

AR can be considered the counterpart of VR, in AR the digital world is the guest in the humans reality and the virtual content overlays onto the real world [14]. In this technology, there is no direct interaction between the digital and the human like in VR or MR being used instead interactions with the device that reproduce some action on the digital content or the use of ML for Object Recognition and Computer Vision (CV) to read Quick Response (QR) Codes or more commonly, ArUco Markers, which the application is able to understand and act on it, like to simulate buttons.

These markers that are added to the environment to aid in registration and tracking are mentioned as fiducials markers [15], there are also others methods instead of using artificial landmarks like using a Simultaneous Localization and Mapping (SLAM) system, which relies in Spatial Mapping and understanding [16] or by making use of Global Positioning System (GPS), while also making using of common inertial sensors integrated in smartphones like accelerometers and gyroscopes. This method has the advantage that works in outdoors environments [15]. AR "immersion" is possible thanks to some features like Tracking [15], Environmental Understanding and Light Estimation.

- **Mixed Reality:** MR is among the most recent terms in XR technologies and while already existing for some years, like Metaverse and Meta (2.2), MR was popularized by Microsoft with the creation of the HoloLens 1 [17], by analysing the Google Trends, this process can be seen starting in 2016 with the public release and being the peak in 2017 when Microsoft allowed the rent of the HoloLens 1 without the full investment of the customers.

  MR looks a lot like AR, and because of AR being already established and well-known, tends to be replaced by it or considered as an extension like Assisted Reality [18], MR consists on a equal interaction between the digital and real world, an AR experience where the human can interact with the digital, preferentially without the need of controllers or secondary

devices, which means, with his own hands like with HL2 or Apple Vision Pro, however to use this technology is necessary a specialized HMD on contrary to AR [19].

When it comes to MR there are two main kind of HMD, the ones like Apple Vision Pro and VIVE XR Elite which makes use of "Passthrough", being in reality opaques, variating mostly in the image quality and grade of color, and the Optical HMD or Optical See-Through (OST) HMD [20] like the HoloLens series and the Google Glass, however Google Glass fits better in the AR category.

- **Spatial Computing:** Spatial Computing is a term defined in 2003 by the Massachusetts Institute of Technology graduate researcher Simon Greenwold, as "human interaction with a machine in which the machine retains and manipulates referents to real objects and spaces" [21], and like many others terms it got popular later, around the announcement of the Apple first XR device in June 2023, the Apple Vision Pro, which while involving technologies like MR and VR such as others HMD like the also upcoming Meta Quest 3, was marketed as a Spatial Computing HMD.

Figure 2.3: Spatial Computing with Apple Vision Pro [22]

During the history of XR others devices were released which also included aspects of Spatial Computing like the Magic Leap 1 or the HoloLens series, whoever these were categorized as

AR/MR devices. With the Apple Vision Pro this category of XR devices also gained terrain in the market, being mainly focused in improving work and in transcending the traditional physical, limited computer to a digital and free dimension.

Spatial Computing is a hard concept to understand since looks a lot like MR/AR, as it can be seen in Figure 2.3, while relaying in VR, this term applies in experiences that reference objects and locations in the physical world, however is more advanced, the digital have a more in depth knowledge of the physical world and of their counterparts in the digital one than in MR/AR.

Still in 2023, in early August, Immersed, a VR desktop application developer company that "was founded under the belief that a future where humanity is able to live wherever/however they want without the constraints of the physical office is fundamentally more freeing than one where we are not." [23] and already known for promoting this mentality in the current existent XR devices thorugh their cross-platform application, announced their own device, the Visor XR, a HMD only targeted for Spatial Computing and work, to improve the use of their application in all three main Operative Systems, however a future release of a Application Programming Interface (API) was declared as possible.

2.1.1 OpenXR and the Cross-Platform Milestone

OpenXR is a standard created by the Khronos Group, first announced in February 2017 and released in March 2019 as a provisional version, it stands as an open-source and royalty-free standard which enables a seamlessly cross-platform development (Figure 2.4), that said with the same base code, the developer can build and release for different XR devices, such as Meta Quest 2 or HTC Vive Pro, till then it was needed to develop different applications for each device, platform by using a SDK provided by the respective manufacturers.

This standard also enabled the development of cross-platform Libraries and Frameworks like the XR Interaction Toolkit from Unity or the MRTK3 from Microsoft. However, while released in 2019, the OpenXR faced some problems surrounding the integration with certain Game Engines and Integrated Development Environment (IDE), like with Unity, being only available starting in

the version 2020.2, released in December 2020, due to some requirements like an Action-Based Input System which Unity was still developing at the time, even some devices which are still not entirely supported by OpenXR, are still being adapted nowadays, an example is the HTC Vive Pro Eye and the OpenXR Eye Gaze Interaction. In the field of XR technologies, OpenXR was an important milestone achieved, but also one that can go unseen.

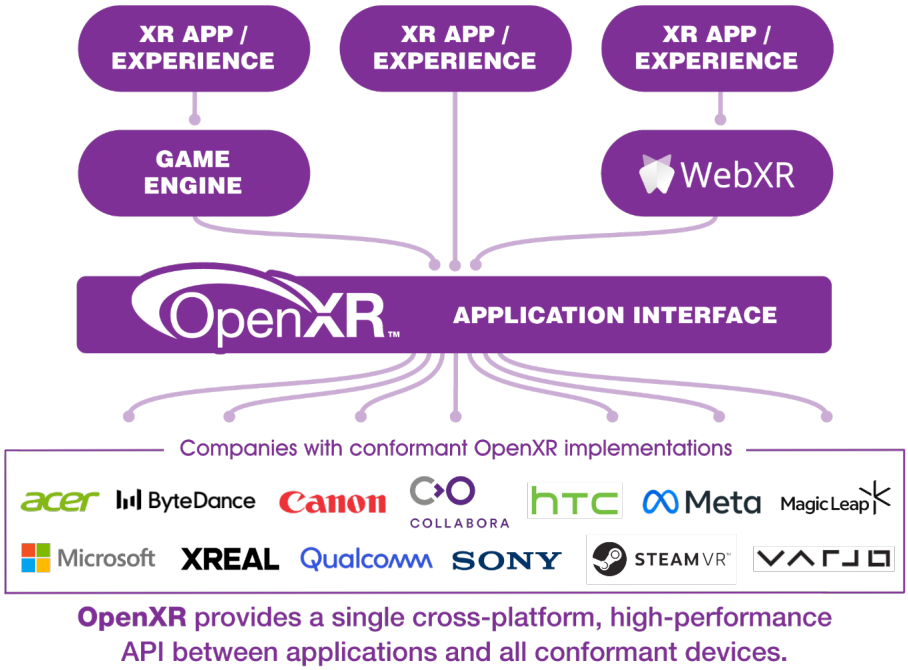

Figure 2.4: "The After" OpenXR [24]

2.2 The Metaverse

The term "Metaverse" was introduced by Neal Stephenson in his book "Snow Crash" in 1992, and it refers to a virtual reality network that acts as a successor to the internet, a digital universe where users can interact, explore virtual environments, and engage in various activities like gaming, socializing, commerce, and communication while offering a shared and immersive space, allowing users to participate from the comfort of their homes.

While the term remained relatively unknown for a long time, it has been featured in games and movies, such as the 2018 film "Ready Player One," which can be considered one of the best

interpretations of the Metaverse.

By checking the Google Trends (Figure 2.5) it can be seen that the term "Metaverse" gained significant attention globally in late 2021, following the name change of the well-known social media company "Facebook" to "Meta Platforms, Inc" (Meta) and her declaration on wanting to create the Metaverse, being already owner of the VR company, Oculus VR, since 2014 [25] and while it can be seen a decline of interest on the next's months, the term Metaverse was never before as well known as it is today, however, misconceptions arose, particularly regarding the use of the term to describe VR multiplayer experiences [26] or life simulator desktop games.

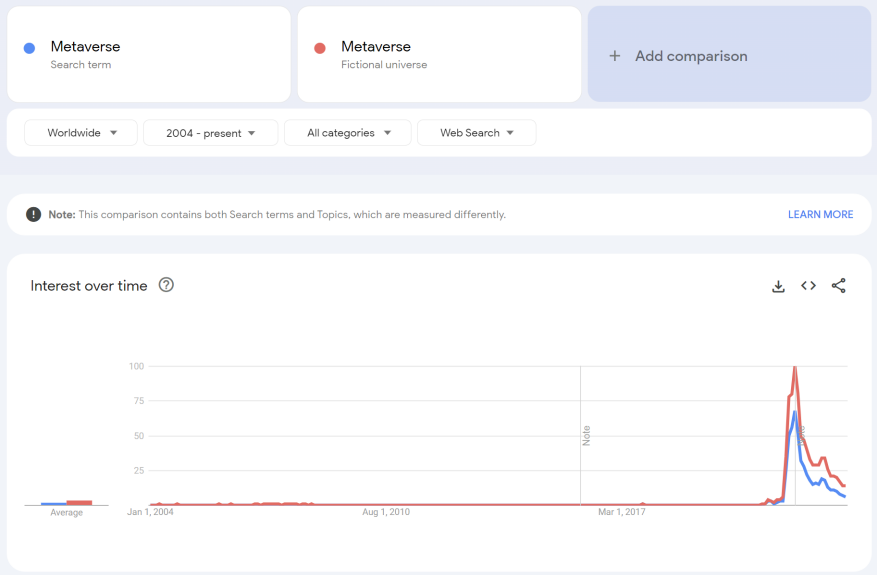

Figure 2.5: Google Trends: Metaverse (Jan 2004 - Sep 2023)

The true essence of the Metaverse involves interactions between individuals using Three-Dimensional (3D) avatars which represent the user not as he is, but as he wants to be, and VR HMD, along with features like common cryptocurrency and seamless transitions between different virtual environments and experiences.

The misuse of the term has created challenges in establishing a clear understanding of the Metaverse. As such its important to note that the project being developed here is not part of the Metaverse and any multiplayer-like features aim to enhance interaction in XR environments and improve healthcare.

2.3 GreenHealth and XR: Enhancing Sustainable Healthcare

GreenHealth refers to the integration of sustainable practices and environmental considerations within the healthcare industry by minimizing the ecological footprint and improving the overall patient well-being.

XR can enable healthcare professionals to do virtual training and simulations, reducing the need for physical resources, reducing waste and providing a secure environment, in the same way it also permits the realization of remote consultations and patient monitoring at distance which reduces the need for patients to travel decreasing the carbon emissions and emotional distress, it also allows the execution of exams or activities which could not be viable otherwise.

In a more advanced approach, while thinking in the next-generation infrastructures (2.5), XR can be taken in account when planning sustainable healthcare facilities thorough the use of holograms for AR and MR which can reduce waste of paper (e.g. patient documentation) or plastic (e.g. support signalization) for example, it can also be used as access to sensors data related to physical spaces, objects and people, minimizing the staff time consumption and improving resources allocation.

Some devices like the HL2 which are standalone and don't use controllers can also be used as portable and intangible computers, minimizing the possibility of equipment contamination and spread while potentiating the access to data remotely and improving the freedom of movement.

2.4 Industry 4.0 and IoT

While the Third Industrial Revolution is attributed to the transition from the mechanical to the digital through the use of computers and gadgets, the Fourth Industrial Revolution happens in the technological world itself, this new stage, also mentioned as Industry 4.0 or 4IR and defined by the founder and executive chairman of the World Economic Forum, Klaus Schwab, was described as "a world where individuals move between digital domains and offline reality with the use of connected technology to enable and manage their lives" [27] and instead of being a transition of ways, is a mix between real and digital, from the use of Machine Learning, IoT, biotechnology or,

previously known as "futuristic" concepts, autonomous cars and Smart Homes [28].

As many others industries, the Healthcare Industry also benefits from the Fourth Industrial Revolution, the use of IoT systems in Healthcare, like observed in Figure 2.6 is named as Internet of Medical Things (IoMT), and among all sectors it can be considered one of the more in need and desperate, with the population growth, the extension of the life expectation rate, the increase of healthcare needs and the inability of the current system to answer all changes [29].

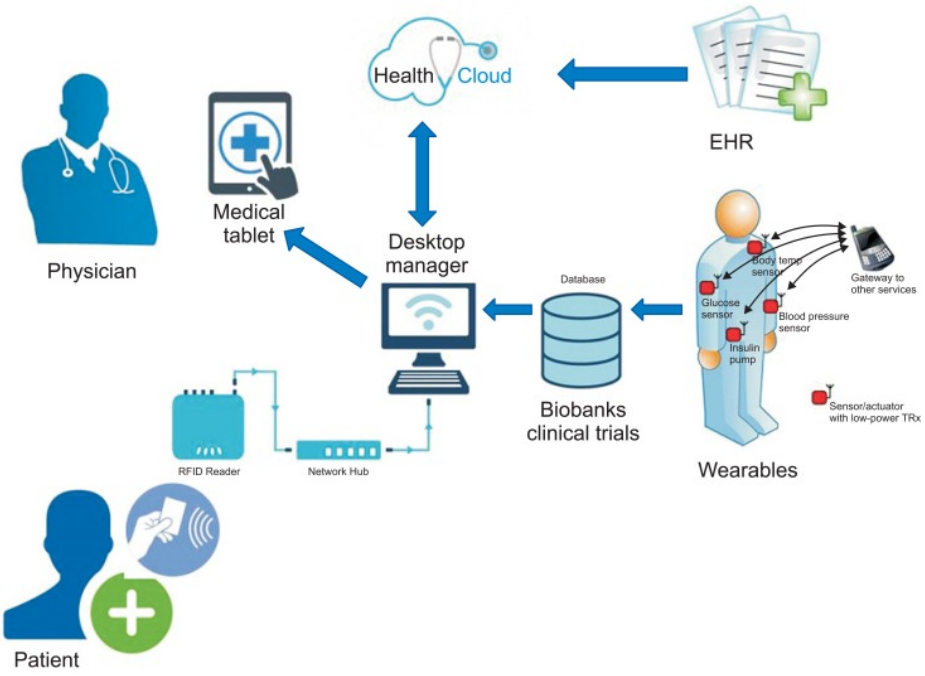

Figure 2.6: Medical Internet of Things in Healthcare [30]

IoMT can improve and enable the care of the patients starting in their own homes, since the use of intelligent toilets to analyze urine or intelligent homes able to adjust noise and lights for autistic patients [31], the use of wearable also consist as an example of IoMT through the broadcasting of real-time biological data which can be better analysed than the default "self-analysis" made by patients while also minimising the need of medical appointments [32].

The use of this devices also enable immediate response on emergencies which could end in fatalities like home accidents with elders, but while able to support the patient in the comfort of their homes, some scenarios still require physical presence in a Hospital such as for operations, however this also can be improved through the use of intelligent ambulances able to do the necessary

diagnostics during the travel accelerating the process [33].

On a more global approach, ecological sensors which could be sending data to some third party systems, can also enable the data to be accessible for Health Systems enabling patients, like asthmatics to be aware of aspects such as pollen and dust level in the air or smart traffic lights connected between them and able to facilitate the travel of ambulances during emergencies [29].

2.5 Next-Generation Infrastructures: Smart Hospital

The term "Smart" have been attributed to a variation of infrastructures, since Smart Homes till Smart Cities, on the middle we can found infrastructures like Smart Hospitals already created like the Cortellucci Vaughan Hospital in Canada or the upcoming Smart Hospital by Valley Health System in New Jersey which will utilize Meditech Expanse as EHR (2.6.1) foundation to implement design elements and technologies that will influence all hospital areas, since network infrastructure to general operations [34].

When thinking in a Smart Hospital, scenarios like doors automatically opening for doctors and nurses in emergencies, monitors sending alerts to nurses when anomalies happens and already notifying the necessary teams for treatments and exams are expected, others scenarios may include position tracking to optimize the healthcare team work or even tracking equipment like wheelchairs or beds [35].

The use of sensors which, for example, allows reading and capture of biometric data, such as movement or heart rate information [36] is also one of the foundations of a Smart Hospital.

2.6 Record Keeping in Healthcare

The healthcare industry is one of the industries that store more data worldwide, being expected the Compound Annual Growth Rate (CAGR) of healthcare data to reach 36% in 2025, part of this due to the growth in the wearables technology that are able to collect medical data [37] such as Wireless Vital Signs Monitoring System (WVSMS).

This amount of data stored digitally is also impressed for ease of use by nurses and doctors due

to the paper-based system used by Health Institutions which is understandable because the need of fast access to data, yet it becomes one of their biggest expenses and makes the industry focused in taking care of humans health part of the reason of the worldwide deforestation [38].

While the progress of technology allowed different ways of accessing health data without the need of paper, this also made that impressed data became less valuable and instead treated as disposable [39], increasing the risk of exposing patients personal data [40].

When it comes to Healthcare digital record keeping methods there is the ones like EHR, Electronic Medical Record (EMR) or Personal Health Record (PHR) which are web-based systems and the ones that are computer-based systems like Picture Archiving And Communication Systems (PACS) which handles medical pictures like X-rays and Magnetic resonance imaging (MRI) or Clinical Document Management Systems (CDMS) that are responsible for managing clinical documents such as medical records, still in the computer-based systems there is also Hospital Information Systems (HIS) which is similar to CDMS, but while CDMS support sharing information between institutions, HIS is directed to be used inside hospital setting [41].

2.6.1 Electronic Health Record

EHR is a e-health tool, which means that uses internet and Information and Communication Technologies (ICT) to share medical information [42], and also is considered essential to the future of recording healthcare data, being one of the necessary components to enable Smart Hospitals and improve healthcare, with interoperability as one of the principles (Figure 2.7).

As a web-based system, with EHR there's no need to install softwares, to worry about maintenance or the device used to access it. It also allows to be accessed from anywhere and depending on how is established the system, without the need of a connection to the Internet.

Although it may appear as a insecure scenario, methods to limit access to web-based systems only in a set of networks does exist and are used [43], and aside from the mentioned points, one of the more advantageous is the centralization of the data, allowing a real-time experience.

In 2015, the National Health Service (NHS) of England setted a goal to be paper-free by 2020, which was changed in his Long Term Plan to 2024, on basis that all services would have met "a

level of core digitalization by 2024" [44], being this digitalization supported by EHR and PHR.

When it comes to EHR and EMR, EHR differentiates by being designed to be shared among different Health Institutions, an inter-organizational system, all the institutions would provide and use information about the patients on a centralized set [45], avoiding situations like repeated or ambiguous data while enabling real-time data and potentializing the use of ML or Data Science (DS), EMR however is limited to be used only within an Institution, privatizing the data.

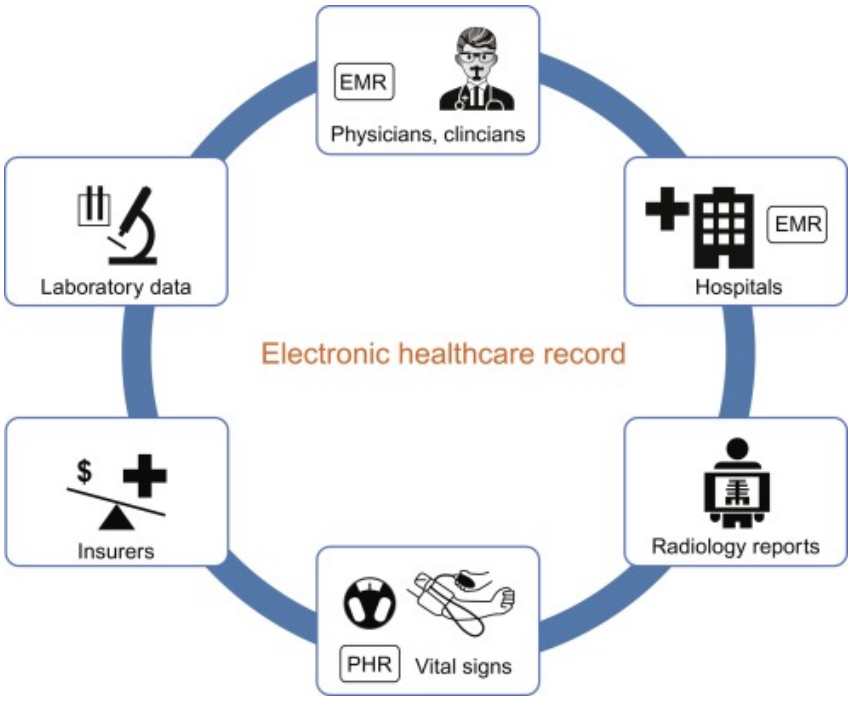

Figure 2.7: Electronic Healthcare Record [46]

While EMR and EHR are designed to be used with different goals, both have as target Health Professionals, instead PHR has a target the patient itself, in PHR the data would be provided and accessed both by the Health Professionals as well from the patients, being he the owner and having full control on his personal data [45].

2.6.2 Health Level 7 and the FHIR standard

The transition from paper-based records to EHR can be expected to be both expensive and complex, from the more technical part, since the system development till the hardware acquisition, as

well as the necessary training of the staff [47], other identified problem is the inability of the already existent EHR solutions to answer the Institutions needs and goals mostly due to the diversity of end-users [48].

When used EHR the medical information don't stay in one organization but is shared, and therefore any information or medical history stored needs to be organized in a way to be understandable by all Institutions and personal involved, which means, standardized, as well to be possible to store already existent data.

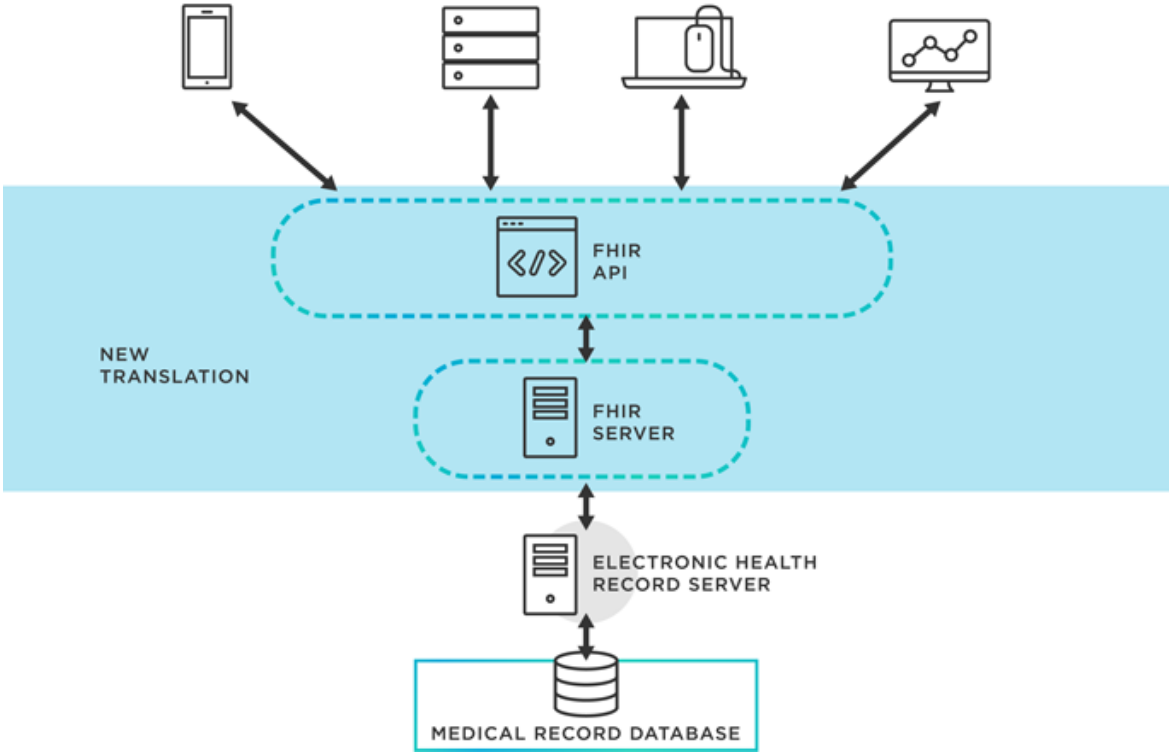

Figure 2.8: FHIR Diagram [49]

HL7, which "7" stands for the Layer 7 (Application Layer) in the Open Systems Interconnection (OSI) model, was founded in 1987 as a nonprofit organization with members from over 50 countries, being accredited by the American National Standards Institute (ANSI) in 1994 [50]. It was founded with the goal of promoting global interoperability in healthcare Information Technology (IT) by creating standards on a multi-year balloting system, and several of their standards like the Clinical Document Architecture (CDA) and Continuity of Care Document (CCD) are referenced as associated with the U.S. Departament of Health and Humans Services program to use

EHR [50].

Fast Healthcare Interoperability Resources (FHIR) is a standard introduced by the organization HL7 in 2011 and first released in 2014 through the version 0.0.82, being resulted from years of development and based on previous HL7 standards, that said, the FHIR standard is still being improved, being the latest major version, the FHIR R5, released in March 2023.

It has the goal to standardize the medical data and exchangeable content through the definition of building blocks referred as "resources", which define the content and structure of information [51], a data exchange syntax and an Application Programming Interface (API) [52], like presented in the Figure 2.8, based on Implementation Guides which assists the development of interoperable health care systems, however theses guides, being mainly consisted on rules, does not specify methods, procedures or tools which stands as demotivating point to participate in such ecosystem [53].

2.7 ArUco Markers

ArUco markers are a type of fiducial markers, these kind of markers were mainly designed to serve as reference points of localization, tracking and pose estimation being crucial for spatial awareness and navigation, they are mostly used in fields such as robotics and CV.

While resembling QR Codes, this markers are more restricted in the encoded data that can contain, being limited to the marker's unique identifier and its position in a coordinate system.

2.8 Barcodes

Barcodes are a technological product of Automatic Identification and Data Capture (AIDC) which is being used from a long time with the goal to contain encoded data for easy retrieval, since the more traditional and common One-Dimensional (1D) barcode, Universal Product Code (UPC) used in retail, till the more famous Two-Dimensional (2D) barcode known as QR Code.

Although QR Codes (Figure 2.9) have great potential, it is mostly poorly used as a "shortcut" or as simple text container, however the AR and MR sector perceives this technology as a way of

improving XR experiences. This technology is also closely related with CV, since mobile devices make use of CV to be able to interpret the Barcodes.

- **QR Code:** Was invented in 1992 by the Japanese company Denso Wave to label automobile parts, is able to store numeric, alphanumeric, binary or kanji data, with different limitations of content size depending of information type, QR Codes are also in constant evolution with the goal to be able to fit more data, being currently in the version 40.

Figure 2.9: Example of ArUco Maker, QR Code and Micro QR Code [Author's Source]

- **Micro QR Code:** Was also invented by the company Denso Wave, being one of the variations created after the success of the QR Code, since based-solutions for different use cases were needed like the FrameQR, iQR Code and the Circular QR Code.

  The Micro QR Code was approved by the Japanese Industrial Standards (JIS) in 2014 and differentiates of the known QR Code by only using one position detection pattern instead of three and needing two-module wide margin, which is half than his predecessor.

  While having a bigger limitation of stored data, for example only 35 numerals, he is more efficient in encoding data than the regular QR Code, which means that his size is hardly affected by the amount of stored data.

  This type of QR Code is also easier to be detected and read when having the same data as the QR Code and can be used on smaller sizes, being currently in the 4th version.

2.9 Artificial Intelligence and Computer Vision Technologies

One of the main technologies that can be used to improve healthcare is ML, which involves and can be found being applied on different fields and for different goals, being some current and famous use cases the use of language models, commonly known as "chatbots", like ChatGPT, natural language processing (e.g. Email filters), synthetic media models (e.g. Deepfake) and Artificial Inteligence (AI) Art. Considering the use of XR as core of the project and taking in account the kind of data available to be retrieved from the XR devices, the use of ML with CV is the pathway with more potential for MR, while VR has more potential for data analysis.

ML is a subset of AI focused in developing models that enable computers to learn based only in data provided by the developer, in a way to be able to make predictions, provide answers without the implementation of the logic and rules, which is created by the computer.

This subset of AI has diverse uses across various fields, including image and speech recognition, along text processing or in creating autonomous vehicles, is being an essential part of areas like manufacturing and healthcare.

CV is the field that enables computers to understand visual information from digital images and videos, being videos a sequence of images, it aims to provide the computer the natural abilities of humans to analyse and process meaningful data, by making use of processes like feature extraction and pattern recognition.

Alone, CV have some limitations like baby humans, while being able to "see" and process the visual data, CV is unable to understand what is seeing, as example we can use CV to track a specific car on a traffic video however the computer will not know that what is tracking is a car, for that it requires the use of ML to be able of doing recognition or more in depth analysis. CV as the potential to improve any industries that relies or makes use of visual information, such as MR and AR devices, or even VR that uses the "Passthrough" technology.

While ML and CV are different concepts, they are commonly used together to achieve some technologies that are nowadays famous worldwide like Face Recognition, but while using ML concepts and methods, Face Recognition or Object Detection are part of the domain of CV.

- **Object Detection:** Is a wide term used to describe the ability of detecting specific objects in images and videos, this kind of models can be trained to detect a variety of objects (Laptop, Table, Chair, ...) or focused on differentiate similar objects (Chair, Wheelchair, Sofa, ...) or even to identify body position like with the game "Rock, Paper, Scissors", depending on the goal and based on the dataset and labels used.

  One of the variations of Object Detection is Face Detection, which normally focus in detecting facial features, such as eyes, nose and mouth to then determine the presence of a face in the image.

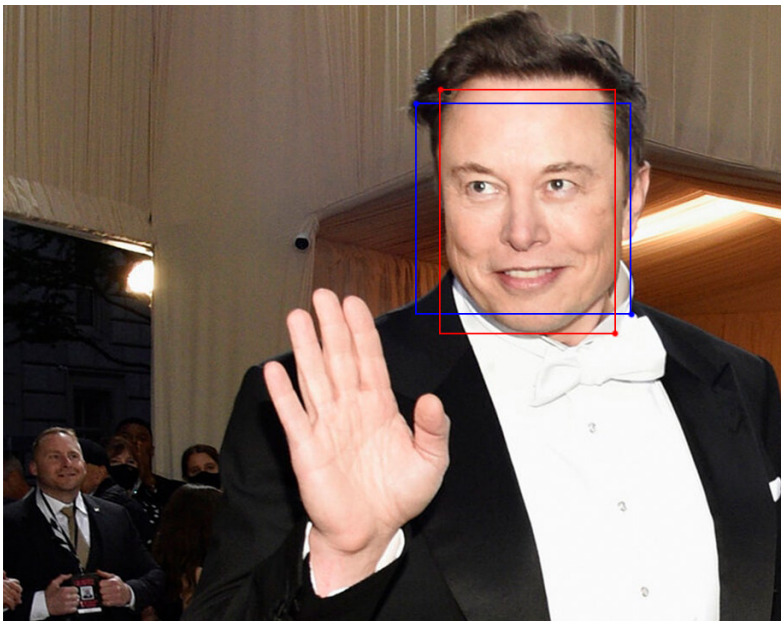

Figure 2.10: Face Detection with Retinaface (Red) and standard OpenCV (Blue) (Adapted from [54])

- **Face Recognition:** Works by calculating the similarity between embeddings, thought methods such as Cosine Similarity or Euclidean Distance, being embeddings the face representation in multi-dimensional vectors. The method used to extract those embeddings, along the quantity and which facial aspects are taken in account is part of what differentiates the models.

  Face Recognition, when used outside of a controlled environment, like in real-time applications, requires the use of Face Detection alongside, being then used the bounding box's

resulted from the Face Detection to extract the embeddings for comparison, the use of different models for Face Detection to mount the Embeddings Database (DB) and for real-time inference may not produce the expected results even if the Face Recognition model was the same due to being provided different bounding box's resulting in different patterns, as shown in Figure 2.10.

- **Emotion Recognition:** Is the ability of detecting the emotion expressed by human, while the emotional state of a human is a difficult thing to divide in labels, when it comes to images, this field normally limits to the more extreme emotions, models trained to do emotion recognition also are commonly focused on the facial expression without taking in account the body language and environment.

  While ML have been progressing in a advance speed, Emotion Recognition is still considered a field with big potential of research and improvement [55], mostly due to biological differences between humans, like skin color or even cultural like for example between the more contained Russians and the more expressive Brazilians.

2.10 Conclusion

One of the main technologies that can be used to improve healthcare is ML, which involves and can be found being applied in a diversity of scenarios, by considering the use of XR as core of the project and taking in account the kind of data available to be retrieved from the XR devices, the use of ML with CV is the pathway with more potential for MR, while VR has more potential for DS and Data Analysis. AR may have equal potential for both technologies.

Chapter 3 State of the Art

In this chapter will be done a review of projects in Extended Reality with the focus in improving Well-Being and Healthcare with special attention to the ones using ML, AI or CV. The projects covered are results found mainly in GitHub, Scopus, PubMed and ResearchGate.

In [56] is developed a project as proof of concept named "Lab Assist Vision", this project results on a MR application made in Unity to be used in HL2 with the goal to assist pharmaceutical laboratories by providing Object Detection, more specifically it was trained a model to recognize laboratory equipment and utensils such as Methanol Bottles and Syringes, tracking these objects afterwards, by combining Azure AI Vision and CV for Object Tracking.

Still related to support applications, [57] proposes a MR application to assist patients in managing their medication pillbox, focusing in the refill moment. This kind of box is commonly used by people that needs to take a large number of pills daily like elders which can also have a hard time in recognising the medication and remember names, however this application recognizes medication box's, and based on the user it overlays the information related to how many should be putted in the pillbox, if any at all.

This functionality was developed to be activated both with gesture or voice commands. When tested with people with different ranges of age it was detected that young people preferred the voice command while the older group preferred the use of gestures, younger people also seemed more comfortable with the use of the device, being used a HL2. While done with goal to be used with this specific HMD, such application could be developed targeting smartphones that support AR which are more affordable and familiar.

Both mentioned applications have different targets, the first to be used by health staff while the second by patients, in [58] is developed a MR application to conduct video calls between doctors or nurses and patients thought the use of PeerConnection with WebRTC, however when used only the view of the user is showed, that is, being used in situations where one side needs more freedom to show details, like body parts or documents, which can be hard with computer camera or smartphones. Still in the same project, the medic is able to verify real-time data from sensors and wearable's being used by the patient in the MR application or in a web application.

This MR application was developed for the old HoloLens 1 and as such some problems like overheating and self shutdown was detected after some time of use, other mentioned problem is the device price, which can not be affordable to the target patient, being him people from rural areas, this can however be changed to use more affordable devices like the upcoming Meta Quest 3 which comes already with the technology "Passthrough" and certain degree of MR, being possible to create a 3D simulation of the doctor space, while this later can use the MR HMD.

FlexiVision [59] is also a MR application for HoloLens 1, it has the goal to assist endoscopy procedures that use Robotic Flexible Endoscope, which are considered better but also hard to do than when with rigid endoscopes. This application projects the view of the endoscope in front of the user by detecting an AR Marker through frustum projection and moves the endoscope camera in accord to the rotation of the HMD, while still enabling the user to manipulate the position and movement with the ladaroscopic gripper of the endoscope with more precision and without the need of checking the screen minimizing neck pain, time consuming and mis-orientation.

It was detected a large decrease of accidental hits, meaning less risks to inflict some internal injury. However while being, in overall, a successfully project the main issue, as with most MR applications, was the learning curve to understand and manipulate the holographic interface, in particular the frustum projection visualization.

One of the features of devices like the HL2 and HoloLens 1 is the Eye Tracking, the ability to detect and track the user gaze behavior, HoloPointer [60] is a "multiplayer" application developed for HoloLens 1 with the Vuforia SDK and relies on the HoloLens spatial tracking functions, it has the goal to be used during laparoscopy surgery training, normally during this surgery the inner body of the patient is showed on a screen and when is needed to transmit instructions or alerts

verbally it can be hard to convey.

This application provides supporting visual information for both, trainer and trainee. After defining the area of the screen in the application through the use of markers meaningful to the SDK, it displays the position of the gaze focus of each other to better comprehension of the situation, overlaying the digital into the physical. Others experiments were done with different kinds of ways to point however most required a free-hand.

In [61] is presented a VR application to promote the execution of Pulmonary Rehabilitation to patients with Chronic Obstructive Pulmonary Disease through exercises executed from their home and guided by a 3D model, during the exercise the patient also wears a Nonin 3150 to measure heart rate and oxygen saturation levels, being the data displayed in real-time inside the VR environment and in a web-based dashboard accessible by the exercise responsible at the hospital.

This research was targeted to elderly people whose benefit more from avoiding going to hospitals and regarding the results the overall opinion was of recommendation and satisfaction, mainly to how easier was to include the exercises in their daily routine, the fact of knowing that those exercise were being monitored by the hospital also improved their confident, reassurance and sense of security. However it was recommended different levels of exercises, regarding difficulty, and also personalised ones.

On contrary to what may seem, XR can also assist blind people, in [62] is presented an AR application that provides navigation support, being mentioned attention to blind people. The application itself provides visual support through arrows and simple signalization for non-blind, however it also gives auditive instructions, such as turn left but also referring the existence of stairs. This is done through the mapping of the local with the smartphone camera sensors and combined with the device motion, while extracting the point cloud result in real time, being also suggested others methods.

Afterwards is defined waypoints nodes and destination nodes, being all data saved on a cloud database. When used by the end-user the application makes download of the map and uses A* pathfinding algorithms to get the best route, different methods to download the map are suggested, like using a QR Code, which would be impracticable for blind people, being also suggested the attribution of identifiers to each map and associate them with a Wi-Fi Service Set Identifier (SSID)

or the GPS location.

When transferred the map, the application will place it over the real-world environment by scanning the environment and finding similar feature points between the view and the mapping data. During the navigation the position of the user is continuously tracked through the use of SLAM tracking feature provided by the libraries ARCore and ARKit.

By using a simple Samsung Gear VR with a Samsung Galaxy S7, [63] uses 360º videos to evaluate the effect of relaxing and nostalgic videos in residents of a nursing home with dementia.

It was tested in thirteen residents (nine female and four male) between the ages 66 and 93 years, being executed in the presence of nurses familiar with the resident and detected a reduction of apathy, a increase of interest and mood improvement through aspects like physical engagement, verbal tone and verbal expression. However one resident showed no interest in the activity and some situations occurred related with cyber sickness like eyestrain.

It was also detected a common interest in scenes which included animals or waterscapes and most of the residents showed interest in reuse VR. Some residents also expressed interest in see scenes related to spaces from their past, one mentioning the hometown, and other the interest in visiting the places where had been stationed during his time in the Defense Force.

While not using a VR HMD, in [64] is presented the project SENSE-GARDEN, in much similar to a project using a CAVE system, this project consist in immersing the patient in a virtual forest through the use of a screen while using sound and smell to better improve the immersion.

This project makes use of a forest environment due to the positive response when it comes to people suffering with dementia [65]. In [66] this project was tested in seven patients with Clinical Dementia Rating 2 through 20 sessions during a period of four weeks being done the assessment at the baseline, after the intervention and lately after eight weeks for comparison, in resume the use of the SENSE-GARDEN showed general improvement on aspects such as cognitive functions, agitation and daily functional level.

However in [67] a comparative study between physical and virtual Calm Rooms is done focused in Psychiatric Inpatients patients, this study involved two different inpatient wards, with a total of 60 patients, since one of the wards already had a physical Calm Room, the patients of that ward, in a total of 20 only executed the session with the physical version, while the other ward

with 40 patients used a VR Calm Room, more specifically an already existent application, Calm Place by Mimmerse.

In overall, this study had also positive results on the use of Calm Rooms, however no big difference was detected between the use of physical or virtual spaces, while may not seem a positive outcome to VR, the use of VR proofs as a acceptable and less costly alternative of building and reserving a unique space for this kind of exercises.

Related to elders suffering with dementia, but with special focus in the ones with Alzheimer, [68] develops a Virtual Reality Driving Simulator as a Serious Game, with the goal to improve Spatial Cognition, this is done by executing sessions of 30 minutes during a period of two weeks, excluding weekends, and was tested with eleven patients.

The tests itself was based in the Morris Water Test, but in a VR environment, and digital clues were implemented like different trees to provide the patient points of reference, each session consisted in four training exercises and one trial, in both the user had 45 seconds to find the target, which is always in the same location. However only in the training exercises it was showed the location of the target at the end of the session if failed, also on contrary to the training exercises, the point from where the patient started in the trial exercises was random between sessions.

The evaluation of the sessions was done by logging the trajectory done by the patient and comparing with expected one, evaluating also the time took, and overall, while the patients were unable to find the target in the expected time interval, it was detected an improvement of time and trajectory done when comparing the data between the baseline and post-intervention.

Different equipment's can also be used together with XR devices to better improve the experience or for different innovative approaches such as in [69] where is developed a VR application to support the integration of patients with Schizophrenia in the society. In this application is simulated a train station where is visible a train, which the patients can enter, and there is realistic Non-Player Characters, also known as NPC, moving in the scenario simulating reality, the patients are able to move by teleportation or walking in real-life, being the movement tracked by the integrated sensors of the HMD, in this case the Meta Quest 2, and reflected in the digital world.

Aside from the VR HMD, this projects also makes use of an Apple Watch Series 6 to track vital signals, being the data saved on DB to further analysis, and, being a main component of this

project, a haptic device, more specifically a bHaptics Tactsuit X40 Haptic Vest, which is able to produce vibrations through the use of 40 Eccentric Rotating Mass motors, being used to simulate collisions of the patient with NPC's or with the scenario props. On a further development it was created a possibility of hugging, using also the bHaptics vests to reproduce the virtual hug in reality.

Still related to Schizophrenia, [70] develops also a VR application where the scenario is personalised to each patient, with the goal to simulate real places like their kitchens. In this application the patient needs to execute a set of tasks based on normal daily life activities, being this managed with AI, with the goal to aim the patient in recovering autonomy.

During the session, data is acquired like Heart Rate and Heart Rate Variation through wearable devices, and is also captured through a external camera being the frames send to inference on ML to emotion and body posture recognition. At the beginning and end of the session the voice of the patient is also captured to analyse his emotional state through the use of Text-Based Emotion Recognition, being the captured voice converted to text. The session and the captured data is also monitored through a dashboard, being the data saved on a DB.

Among the constraints of the daily well-being of a person, one which may pass as laughable, can be the limitations imposed by their phobias, the extreme fear of something, since the more livable like Cynophobia, the fear of dogs, till the more aggressive ones such as Mysophobia, the fear of germs, which are referred as Obsessive-Compulsive Disorder (OCD). While some people accept to live with those phobias, some go under treatments to overcome them, normally these treatments include Exposure Therapy [71], or Cognitive Behavioral Therapy (CTB), which implies direct interaction with the target of their phobias, however some may not be as easy as to pat a dog, in [72] is developed a VR application to support the treatment of Squalophobia, the fear of Sharks, which my constrain a person in swimming or doing aquatic recreational activities, depending on the grade of phobia, this may even make a person lose sense of logic and avoid places like swimming pools or provoke real incidents due to the possibility of see any unknown subaquatic object as a shark.

This treatment is done by using a procedure known as Virtual Reality Exposure Therapy (VRET) and it was tested on a single 30-year-old Caucasian woman who contacted the research

team when knew about their development of different VRET through the media. Three scenarios were developed, a pool, a lake and a sea, all scenarios had different grades of depth and dark areas, it was also implemented related objects like normal fish's and sea plants with their own animations and behavior, a shark would also appeared or disappeared when toggled by the therapist.

Through the sessions it was detected an improvement of the patient, starting from being more relaxed when seeing unknown objects till not panicking when seen a shark from afar, the patient was also consulted again after a year, being related that she no more feared swimming alone. While being only tested in one person, this application showed positive results, still it may reach a limit on how much can improve the state of the person since in this project it was used a Sony PSVR HMD which are not waterproof, like most of XR devices. However if used the waterproof DIVR® from BallastVR the patient can be submerged during the session which would greatly improve the immersion, this HMD is similar to Samsung's Gear VR, which means that requires the use of a smartphone.

3.1 Conclusion

XR technologies can be defined as a topic of interest in the research field, having being studied it use for different goals as was observed, if removed the focus in improving well-being and healthcare even more interesting and innovative approaches can be discovered, such as in industry, tourism or marketing, however as can be expected most off the studies end in small, contained applications, some even with the main concept repeated, which if introduced in the real workflow could end in causing confusion and not behave as expected.

Chapter 4 Conceptual Model

This chapter has as goal the conceptualization of a system focused in the use of Extended Reality and Machine Learning to improve healthcare and empower institutions, making used of interoperability, the proposed components should not work as solo services/applications but as a whole system focused on the same goal.

During the review of the State of the Art it was seen the appliance of XR, alone or with others technologies, like ML, or even others devices like sensors, to answer current problems or to empower our Health Institutions. However, there is a lack of interoperability, a concept present in the future of healthcare, just as there is a lack of research and projects in that direction.

Interoperability refers to the ability of different systems, software, or components to work together effectively, enabling seamless communication and interaction. It ensures that diverse technologies can exchange data and functionality, fostering compatibility and cooperation across a variety of platforms and environments.

Others concepts were also identified as part of that future such as EHR, Smart Hospital (SH), IoT and standards for data interoperability like FHIR. Technologies like ML, CV and Data Science also stand as crucial aspects, being involved directly or indirectly. However such developments also introduce interrogations about adjacent implications like the environmental effect, from which terms like GreenHealth appear.

The developed conceptual model (Figure 4.1) comes as proposal of developing solutions which aim to that future with focus in the use of XR technologies, which shows potential in being eco-friendly [73], having as major constraint the user adaptability.

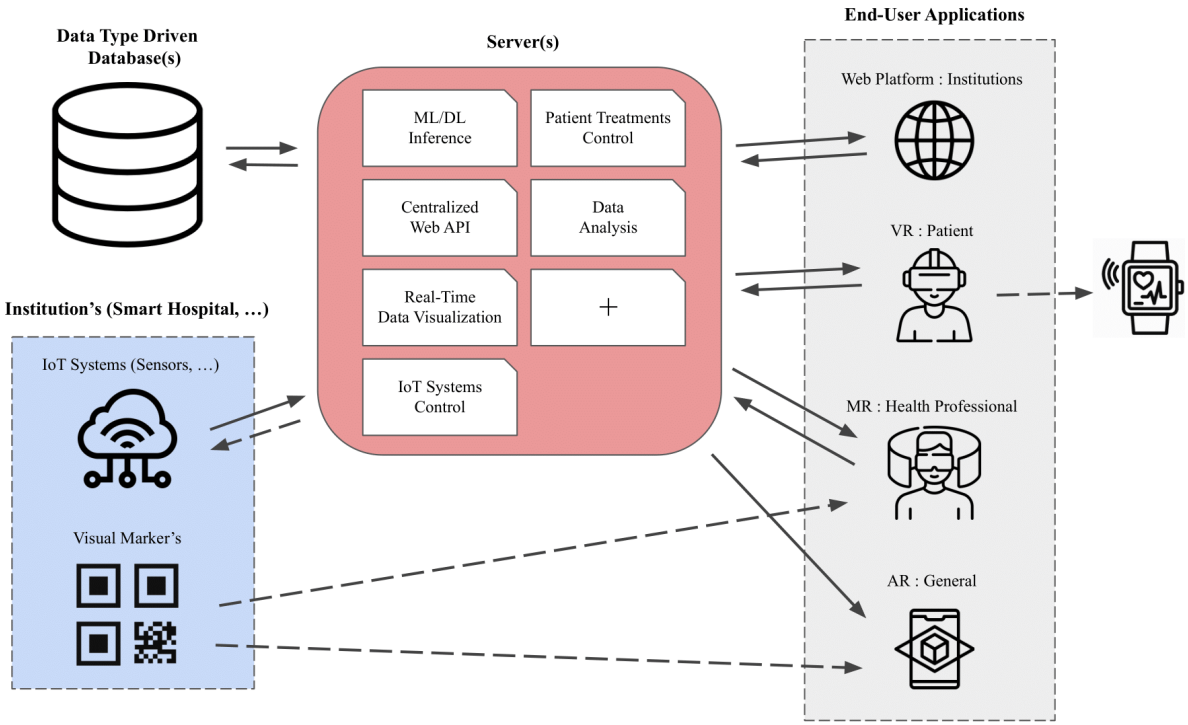

Figure 4.1: Conceptual Model [Author's Source]

- **Data Type Driven Databases:** Between basic hospital data such as patient identification and historic, IoT data and Big Data, a unique DB can not be able to answer all components or even if able, not be optimized for each type of data or services, as such the chosen DB's must be based on the type of data being saved.

- **Institutions Components:** When it comes to Institutions, being (Smart) Hospitals, Nursing Homes or Medical Centers, between others, these can be equipped with different IoT devices such as heating/cooling systems or Smart TV's, being one of the principals and acknowledge, the sensors, which also presents themselves in the more different forms such as temperature sensor or proximity, even utensils can include sensors such as diapers.

  For the conceptualization of this model was taken in account the available RAK4630 (Figure 4.2), a sensor which even with a base is no bigger than 6cm x 3cm and can be found selling for less than 50€, it also supports modules to detect aspects like humidity, temperature and atmospheric pressure, all of which are important data for maintaining patient's well-being.

The mentioned sensor and many alike can be connected to a LoRaWAN Gateway and broadcast data in real-time through LoRaWAN protocol, each Gateway able to support up to ten thousand devices within a range of 15 kilometers (in rural areas), the gateways are connected between them and the endpoint of the data can be defined in the dashboard of The Things Networks, being the data encrypted and secured.

Figure 4.2: RAK4630 connected to Powerbank Xiaomi Redmi 2 20000mAh 18W [Author's Source]

The mentioned sensor just need to be connected to a powerbank and may stay active during weeks without needing recharge, aside compatible devices, there are others specifically created for the LoRaWan protocol like the LW001-BG LoRaWAN GPS, a GPS tracker which can be used for example to track wheelchairs or ambulances, this means that if coverage in the same or different gateways, an ambulance can be tracked between different institutions or even the whole city. Other examples can be the use of beacons such as the iBKS Plus which can detect and seamless contact with XR HMD.

Still all mentioned data is normally only visible in computer/mobile applications or web applications, however by populating the Institution with Visual Markers such as QR Codes with identifiers interpreted by a hand-free MR application, for example, in a summer night, in a hallway connected to patients rooms filled with the mentioned sensors, the nurse can

just walk by and through the interpretation of the code create a virtual panel where the data can be seen, this however is not limited to that, the same code may be associated to the room itself and enable a panel where is listed the patients inside the room and so much more. The same applies to AR applications being also possible to filter what is seen through the same QR Code depending on the logged account.

- **Institution Web Application:** While XR HMD are optimal to data visualization and consult, the same does not apply to inserting data, the use of controllers make the process slow and hand tracking is still evolving, even the more advanced HMD may not able to follow the same hand writing speed used when in physical keyboards.

  Aside of that there's also some expected limitations, firstly related to the number of devices owned by the institution, the battery lifetime and the insertion of data such as images or files. In the same way some features are also more feasible to be done through a physical device like remote systems controlling;

- **Patients and Virtual Reality:** VR HMD are among the more financially accessible XR devices, having a big variety on quality and hardware depending on the person goals, among them, the more famous one is the Meta Quest 2, formerly known as Oculus Quest 2, this comes for is low price hardly beaten in the market, around 283 euros in 2023, and for being standalone per default which makes this device the one normally bought for first-timers, and while expected to be replaced by the released Meta Quest 3, this one also has a competitive price around 473 euros on release.

  VR also works better and have more potential when used by and in patients as it can be seen in the State of the Art, the ability to fully surround the person in digital scenarios enables the execution of treatments or procedures at distance while keeping the patient in their comfort zone promoting security and well-being, which indirectly also benefits others aspects like expenses on physical material, hospital overcrowding [74], hospital staff overwork [75] and even environmental aspects such as lowering the need of long travels to the big hospitals, diminishing pollution and roads traffic [76].

- **Health Professional and Mixed Reality:** While VR has great benefits for the patient side, the same may not apply to the health professional, these have a need of keeping their hands free to execute their tasks which makes the need of using controllers a obstacle, being surrounded by digital worlds also does not present benefits for them since they need to be free to move in and between physical spaces as well be able to communicate between them.

  These and others factors, make the use of MR, or AR HMD, the ones who better can improve the health professional work, the same is also justified by the State of the Art, where most of the research done to improve the doctors work is done by using HL2.

  MR presents possibilities like Spatial Computing this is, the ability to bring their computers with them, remote collaboration [77] or data visualization through holograms [78], all of these with a intangible, hand-free and contamination-free approach;

- **General Access through Augmented Reality:** AR devices are the more common and accessible for everyone, each year the ability to support AR becomes a must have in the phones or tables, and while most of the current generation of elders are still not comfortable with such devices, the next generations have grown with this technology.

  AR can be used by anyone, Health Professional which can use instead of MR if to only visualize data as well it can possible for the Patient, and Civilians, being patient familiars, new patients or even external staff like janitors or electricians can use AR for basic things such as to guide himself through the Hospital or also interpret QR Codes and, as it is more common, ArUco's.

Chapter 5 Hardware, Software and Technologies

This chapter is dedicated to describe the hardware, software and technologies used during the prototypes development and chosen taking in account the conceptual model defined on the previous chapter for further developments.

5.1 Equipment

Two XR devices (Figure 5.1), one for VR and the other for MR, were used for the project. These devices were provided by UNIAG thanks to the funds of the GreenHealth Project and are currently located at EsACT in Mirandela. In terms of AR, at the time of the development of the prototypes it was made known the incompatibility of the available equipment and was not possible to acquire a new one. Additionally, two servers were used to host the project's back-end component, a Cloud-Based Server and, due to the limitations of the Cloud Server provider free tier, a On-Premise Server.

- **HoloLens 2:** The HL2 is the continuation and improvement of the HoloLens 1, the world's first fully untethered holographic computer, and its a MR HMD created by Microsoft and released in 2019, it works as standalone device without the need of controllers, making use of hand tracking to manipulate the interface. In contrary of others XR devices which promote the inclusion of the MR, HL2 does not make use of the technology known as "Passthrough", being this the case of others devices like the Meta Quest Pro or the upcoming Apple Vision Pro.

The term "Passthrough" refers to a technology that can be considered a mixture of VR and AR features to provide a MR experience, and on contrary of devices which are more fit in MR category, like the Magic Leap 2, the HL2 have both the battery and the Central Processing Unit (CPU) integrated on the HMD instead of being external, which makes the HL2 one of the few HMD truly standalone and providers of a MR experience, aside of Hand Tracking, the HL2 also makes use of Eye Tracking, a necessary technology when not using "Passthrough", and which makes necessary to do Eye Calibration, ideally for each device user, being possible to have *N* number of users/accounts.

Figure 5.1: HTC Vive Pro Eye and HoloLens 2 [Author's Source]

However the HL2 hardware can be considered outdated when compared with his competitors due to the fast evolution in the XR World in the lasts years, another problem is the price which, while fair to the HMD, affects his popularity and by consequence its integration on enterprises, which are the goal of Microsoft instead of being for entertainment [79]. For first time users the HL2 can also end in some disappointments, while improved when compared with it antecessor, the current Field of View (FoV), the actual part of the glasses in which is displayed digital content, is still small considering the expected and affects the usability.

While a HoloLens 3, with a hardware improvement and a expected increase of the FoV, could become a revolution on the MR market, the HoloLens future have been uncertain due to Microsoft continuous change of position related to his development, being in great part dependent in the end result of the contract between Microsoft and the United States Army

of $22 billion dollars, a deal signed during the year 2021 for the development and supply of military grade HoloLens in a period of 10 years.

- **HTC Vive Pro Eye:** The HTC Vive Pro Eye is a wired VR HMD, also known as tethered headset, manufactured by the HTC Corporation and released in 2019, and in many ways is similar with the HTC Vive Pro, released in 2018 which was an upgrade of the first HTC Vive VR HMD, the HTC Vive from 2016, being the main difference the use of built-in Eye Tracking by using Tobii XR, this was also the first VR device integrating this technology.

  While being naturally a wired HMD its possible to use a device from the HTC Vive to enable a wireless connection, however both scenarios together with the fact that the HTC Vive Pro requires a powerfull gaming desktop with a DisplayPort entry and his price has been a constraint for his popularization in comparison with others VR HMD like the Meta Quest, aside from that the HTC Vive Pro can be considered a good option for situations where quality and performance has more priority than portability.

- **On-Premise Server:** Is a designation used for servers that are controlled and maintained by the company that use them, which means that the company is in charge of refrigeration, upgrades, reparations, security and others aspects that are necessary to take in account when maintaining a server.

- **Cloud-Based Server:** Another term to designate a server aside from On-Premise is Cloud-Based, more commonly know as just Cloud, which can be resumed as a On-Premise Server which is not being maintaining by those who use it.

  The use of Cloud-Based Server are being increased on the last years thanks to providers like Amazon, Oracle and Microsoft, each one with is own group of offers. By using a Cloud-Based Server or Service, the user, being a company or a single person does not need to worry about mounting and keeping the hardware which can become costly and require specific IT skills.

  Cloud-Based Server is part of the concept Cloud Computing, whoever Cloud Computing comes in different flavours, Amazon Web Services (AWS) as example, divides their services

between Infrastructure as a Service (IaaS), Software as a Service (PaaS) and Platform as a Service (SaaS), which can be resume as how much control has the user over the actual hardware.

5.2 Development Support

In this section will be presented the tools that supported the development of the prototypes for a development environment understanding and tools suggestions to the reader.

- **Development Environment:** IDE are softwares used for development which provides the necessary tools and UX to optimize the development for a certain language or field, depending on the IDE, normally associated with compiled programming languages like C#, there are also IDE for interpreted languages like PyCharm, for Python.

  It was used the Visual Studio Community, a free IDE by Microsoft that provides support for the development with programming languages and in fields of action of them, like Web Delopment with ASP.NET, Universal Windows Platform (UWP) applications, C#, C++, between others. As necessary for some components developed it was also used the Intellij IDEA, an IDE by JetBrains, same creator of PyCharm, and is focused on development for desktop using Java and Kotlin, but also offers support for others programming languages that make use of Java Virtual Machine (JVM), while Java is little used in the project, due to being a compiled language it was necessary the use of an IDE.

  Aside from the mentioned IDE's it was used the Source Code Editor Visual Studio Code, also known as VS Code, created by Microsoft, it provides the necessary tools to code in many languages and manipulate files while keeping simplicity, being also possible to use extensions to empower him in diverse ways, VS Code was one of the main tools used for development, to write code for Python and Go among others minor uses.

- **VMware Workstation Pro:** VMWare Workstation Pro is a hosted (Type 2) hypervisor that runs on x64 versions of Windows and on Linux operating systems, like VirtualBox, another well known hypervisor, its used to create Virtual Machine (VM) of operating systems,

being them Apple, Microsoft or Linux based systems. VMWare Workstation Pro is used to simulate the on-premise server and it was chosen instead of the free VirtualBox due to performance issues.

- **MobaXterm:** Is a Windows application used for remote computing, it allows the use of Linux commands and provides access to machines through the most used remote network tools protocols as Secure Shell (SSH), File Transfer Protocol (FTP), Telnet and others, while also possible to write and synchronize files by using a code editor without complications or even upload files with simple drag-and-drop, being also provided visual information of the file hierarchy.

  This tool was used to manage the servers and develop directly on them, in the beginning of development it was used the Visual Studio Code with the plugin Remote SSH, however the connection with the Cloud-Based Server was unstable.

- **Postman:** Is a platform used by developers to support the development of Web API's, covering most of the necessities that a developer may have, since the ability to test most of types Hypertext Transfer Protocol (HTTP) requests, GraphQL, Socket.IO, WebSockets (WS) and, most recently, Google Remote Procedure Call (gRPC) services.

  The developer can save existent endpoints, with a example of a request and response, as well group them in collections, being possible to create teams and work simultaneously in the same project, enabling faster and cleaner communication between back-end developers and front-end developers.

- **Unity:** Unity Engine or Unity3D is among the most known and used Game Engines, it provides powerful tools and packages to develop games while also having an online store, Unity Asset Store, where can be found a variety of assets to support and empower the engine.

  While there is a variety of Game Engines or specific tools created for XR development, like the game engine Unreal or the open-source C# library StereoKit, Unity has being established on the XR market as the primary choice, both in the opinion of the developer's

community as well the technology pioneers, due to his great support, ease of use and focus on improving XR development, from the integration of the XR development standard OpenXR and the creation and continuous improvement of the library XR Interaction Toolkit and AR Foundation.

- **Trello:** Is a project management web platform mainly used for team projects, being the main page a lot like a Kanban board [80], it gives a lot of freedom of organization and can be empowered with the use of add-ons, which are called Power-Ups, it also have other features like automation to optimize performance.

  While this thesis/project is a one person team, without previous work done and no expected need of work combination with others persons, this projects involves a lot of different components, different technologies and learning process, being used then as self-organization and also to save related research.

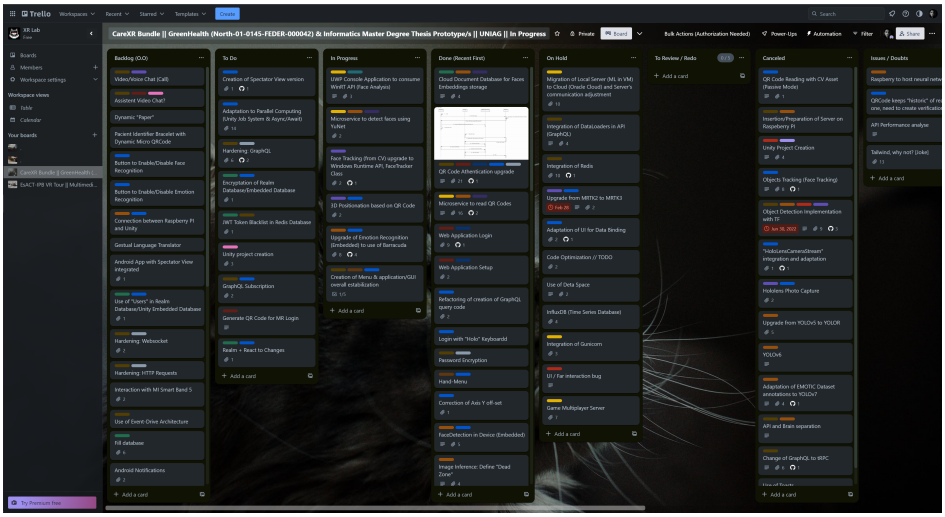

Figure 5.2: Trello Project Board [Author's Source]

While not being used a established standard for organization it was mainly divided in five lists "Backlog", "To Do", "In Progress", "Done" and "In Hold", being used others lists to store general data or annotations, Trello was also used to provide support in future projects involving XR development (Figure 5.2).

- **Git and GitHub:** Git is a distributed version control system used by developers to track

project development and versions, both in solo and team projects. While primarily used through the command line, there are also Graphic User Interface (GUI) applications available to simplify the process.

For this project, GitHub Desktop, developed by GitHub, was chosen as the GUI application to be used on the development computers, while the servers used only the console, a total of 21 branches were used (Table 5.1), excluding the 'main' branch, which remained untouched.

Table 5.1: Git Branchs Structure

Branch Name	**Nomenclature**
dev-UWP_FaceDetector	dev = Development
dev-MS_FaceDetection_csharp	test = Test
dev-MS_FaceRecognition	obs = Obsolete / Ex-Dev
obs-MS_FaceRecognition	bck = Backup
dev-MS_ImageInference	utils = Utility
dev-UP_MixedReality	UP = Unity Project
dev-API_WebSockets	DB = Database
dev-MS_QRCode	MS = Microservice (MS)
bck-Nginx	WS = Website
dev-MS_PersonDetection	UWP = Universal Windows Platform
dev-WS_ReactJS_MUI	API = Web API
dev-API_Go	
test-API_FastAPI_Clean	
dev-MS_EmotionRecognition	
obs-WS_OutSystems	
utils-ProtoCsharpCompiler	
bck-DB_Neo4j	
obs-API_FastAPI	
test-API_AspNetCore	
dev-UP_AugmentedReality	
dev-UP_VirtualReality	

- **Adobe Illustrator:** Adobe Illustrator, more commenly called as Illustrator or Ai, is a vector design tool used when vectorization of image is a necessary requirement, like for Illustrations or UI, in this project the Illustrator is used as tool to design the interface for the MR application which requires to be in vectors for a better renderization.

5.3 Programming Languages

In this section will be mentioned and given a little description of the programming languages used during the development and referenced where they are present.

- **C#/C-Sharp:** Also known as C-Sharp, is an Object-Oriented Programming (OOP) language created by Microsoft and had it first version released in 2001. It has its roots in the C language and is often considered the Microsoft counterpart of Java. C# is a compiled and type-safe language with many use cases, such as game and web development, due to its ability to run on the .NET framework. In this project, C# is used in the XR applications developed due to the use of Unity Game Engine.

- **Go/Golang:** Go, formerly known as Golang, is a programming language developed by Google since 2007 and released in 2012. It is a compiled language that emphasizes structured programming instead of being OOP, however some features of OOP can be replicated through structures. It incorporates features from other languages like Python and C while focusing in simplicity.

  Go is gaining popularity, especially in back-end and cloud development, due to its small executable size, performance and execution speed, it was also designed to run on multiple cores, to enable concurrency in a simplified way and to scale as cores are added. However, due to his simplicity it comes with problem of taking more time when developing and can be considered a hard language to learn base on the already known by the developer. In this project, Go is utilized to build the main Web API (6.4.1) which runs on a Cloud-Based server (5.1).

- **JavaScript:** Is a worldwide famous programming language, due to its extensive presence in web development, encompassing both front-end and back-end. It is utilized in numerous libraries and frameworks within this field, including React.js, Vue.js and even Bootstrap. It also has a strong presence in others fields such as back-end with the server environment Node.js and even mobile development with the cross-platform framework React Native.

While not strictly an OOP language, JavaScript does support that programming methodology.

While being a widely adopted language, considered the "per default" when it comes to web development, it does have its drawbacks, such as being dynamically typed only. Several libraries and programming languages like TypeScript have emerged to address these issues or even serve as substitutes, whoever, similar to Java, JavaScript has already become an indispensable language. In this project, JavaScript is primarily used for WS development and the web application prototype through the use of React.js.

- **Python:** A well-known programming language that was released in 1991, since then it has become a standard language for beginners who want to learn programming and avoid more complex languages like C and C++ due to its low learning curve. Is considered a high-level programming language, which means its syntax closely resembles human language. These factors, along with a large and supportive community that continuously improves the language through packages, like Numpy for scientific computing or Pandas for data analysis, have made Python one of the main languages used in fields such as ML, AI, and DS which are inherently complex, and which many ML developers and researchers come from mathematical backgrounds rather than programming. Consequently, Python is utilized in the project components that involve ML in the back-end (6.4.2).

- **C++:** Also know as CPP, is a programming language released in 1985 as an extension of the C programming language with the goal to add new features that were being became mainstream and present on newer programming languages like OOP and generics. In this project, C++ is only present on one specific MS which uses a library for Face Detection in C++.

- **Java:** Is a OOP class-based programming language released on 1995, while it had the goal to be a general-purpose programming language, she is more known as a Android Development programming language and for server-side development, in this project she is used on a MS to detect and read QR Codes from images.

5.4 Databases

In this section it will be presented types of DB present on the conceptualization of the project, both embedded and client-server ones, further specification related to the DB provider will be exposed in the Prototypes Implementation chapter (6.2).

When it comes to DB there are two main uses, as Embedded DB or Client-Server DB, the first is used by an application to store data internally, for data persistence, normally only stores temporary data and in low quantity, however can also be used to store larger volumes of data if not to be expected a continuous consultation of the Client-Server DB, one of the most common uses are on mobile applications, like Facebook or Instagram, in which is stored some degree of content to be able to be seen without the need of connection to the Internet, this kind of DB normally don't store files such as Portable Document Format (PDF) or images, on the other hand, Client-Server DB refer to a DB running on a server, which will store big amounts of data and can be accessed by anyone, given the necessary credentials of access.

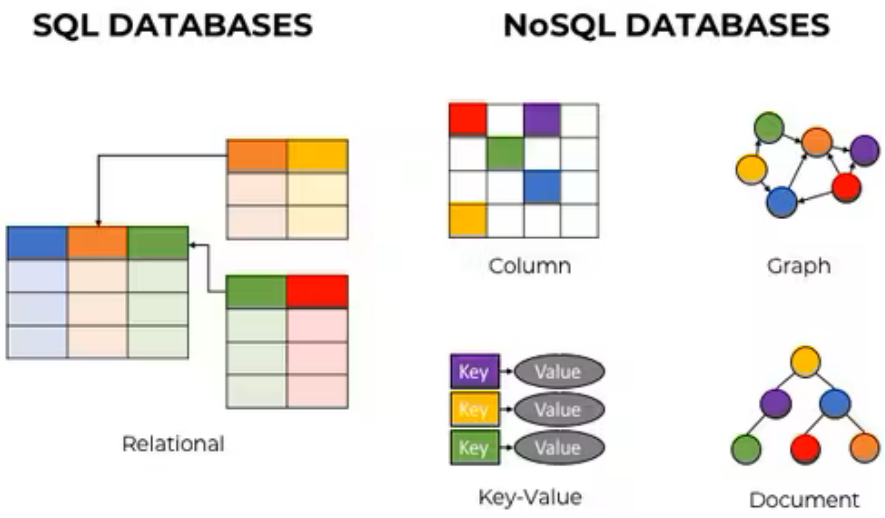

Figure 5.3: SQL Databases and examples of NoSQL Databases [81]

Aside from Embedded and Client-Server DB, other way to classify DB is between Structured Query Language (SQL) and Not Only SQL (NoSQL) DB (Figure 5.3), among the two type, SQL DB are the more commonly used and normally it refers directly to Relational DB, a type of DB that makes use of integrity constraints, data structuring [82] and which uses SQL as norm for making

Create, Read, Update, Delete (CRUD) operations with some variations between DB providers (Oracle, MariaDB, MySQL, ...), being the two terms used together to refer to structured DB that relay on relationships between the data stored [83]. While NoSQL refers to DB that do not use SQL on his queries, also on contrary to SQL DB that can be resumed to Relation DB, NoSQL DB is divided in more than one type of DB, such as Document DB and Time-Series DB [83].

NoSQL DB surged as response to the increase volume of data, different types of data and Big Data, being the later a term heavily connected to Data Analyse, which is the process of analysing large volumes of data to get insights such as patterns, market trends or consumer preferences, since SQL DB has shown efficiency problems and problematic horizontal scaling, companies worldwide have starting considering NoSQL DB [84] and even while SQL DB are preferable for vertical scaling, the costs involved are exceptionally increasing [85].

As per 2023, the official NoSQL website defines 15 categories of DB [86] but it can be expected the creation of new NoSQL DB, as is the propagated ideology of the NoSQL community, "Use the right tool for the job" [87]. In this section will be described the categories which are used on the conceptual model and in the prototypes development.

- **Graph Database:** Based on the graph data structure and popularized around the year 2000, Graph DB makes use of Nodes (Vertices) and Relationships (Edges) to store meaningful data, on contrary of Relation DB, this kind of DB is able to store data and define labels in the relationships [88], it also provides better navigation and search mechanism through the relationships without the need of JOINS by using traversal algorithms instead, Graph DB are also better handlers of relations many-to-many [87].

  By being based on the graph data structure, Graph DB have already an advantage to create Recommendation Systems, Decision-Make Support Systems or in making use of DS algorithms like Link Analysis. Due to these points and how heterogeneous Healthcare data can be, a Graph DB is used as the system primary DB.

- **Document Store Database:** Document Store DB, or Document DB, are considered the "per default" of NoSQL DB, being the term forgotten in many occasions, Documet DB offer great degree of freedom when saving data which is commonly saved on JavaScript Object

Notation (JSON) documents which are grouped in collections, the equivalent of tables in Relational DB, without requiring the definition of a constant schema as is required on SQL DB [84].

Aside from not using SQL for queries execution, Document DB shines when it comes to store Big Data, and while its possible to make use of relationships, Document DB are more oriented and fit to store unrelational data [89]. In this project, Document DB is used to store that kind of data, with the goal of complementing the Graph DB, from QR Code content till VR Hotspots.

- **Object Database:** Also known as Object-Oriented or Object-Based DB, is an old type of DB that incorporates OOP concepts such as inheritance or custom data types [90], together with Relational DB, both surged on the decade of 1970, but while Relational DB became a industry standard, her partner was putted forgotten, only in the last decades, with the growing of the NoSQL DB and OOP is that the Object-Oriented DB started to get some attention. While still hard to be founded being used as Client-Server DB, by being a type of DB that shares the same concepts as OOP Languages, shows potential to be used as an Embedded DB on applications OOP, being this the use case of this type of DB in this project.

- **Time-Series Database:** Other type of NoSQL DB are the Time-Series DB, these kind of DB are not new, and they may look like Relational DB on the surface with a fixed column for timestamp in most DB providers, while on the behind, this DB are optimized to manipulate time related data and receive constantly new data, however their use was mostly for financial data, stock trading and trading for example [91].

  Only nowadays with the propagation of the concept of IoT and the Fourth Industrial Revolution, named as Industry 4.0, time-series data and metrics are constantly being streamed by devices like sensors, requiring a DB that can handle, while possible to find use cases of Document DB or even Column Store DB acting as Time-Series with satisfying results [92], Time-Series DB was created specifically to handle these kind of scenarios [93].

- **Key-Value Database:** This type of DB differs greatly from other DB, starting from being mainly in-memory instead of on-disk, which means that the data, normally temporary, is stored in the Random Access Memory (RAM) instead of a hard-disk and while not being a limitation, this is the common approach.

  This kind of DB are designed to optimize high-speed data access while providing simplicity and scalability, in terms of structure, Key-Value DB resemble the data structure Dictionary [87]. While having many potential uses cases, this DB is used to support horizontal scalability of the WS services.

5.5 Network Communication

When choosing the technologies to be used on the network communication it was taken in account the end-user devices, the XR HMD, and the necessity of optimal communication.

- **GraphQL:** A query language for API created by Facebook as a substitute of Representational State Transfer (REST) API to be used by their mobile applications and to solve some known problems of REST API like Over-Fetching and Under-Fetching which become more problematic on mobile applications, having as concept "Ask for what you need, get exactly that", it started to be used in production by 2012, and is open source since 2015.

  GraphQL shares some of the concepts of REST like both using JSON as default data serialization, but GraphQL only makes use of one POST route in which will be provided the query, in the request body, for the API based on a pre-defined schema in which is defined all kind of data available and how, the normal concept of CRUD operations also are divided on Queries (Read) and Mutations (Create, Update, Delete) [94].

  When compared with Remote Procedure Call (RPC) or REST/Simple Object Access Protocol (SOAP), GraphQL provides more independence between the front-end and back-end developer. While solving some problems of REST, it also bringed new ones, by being based on a POST request, GraphQL does not make use of cache and due to being based on graphs, when used with most kind of DB it causes a problem known as Problem N+1 [95], however

by being used a Graph DB as primary DB, this problems is hardly noticeable. Other identified problem is that while GraphQL optimizes the communication client-server by only returning what requested, on most cases the server will process all the data referred on the schema even if not requested, this can worsen the Problem N+1 and prejudice performance if used slow languages. An important aspect of GraphQL that should be understand is that it is not to be used as an indirect method for the Front-End to query DB.

- **gRPC:** Is a Buffer Protocol, considered the new age of RPC, and while not official, "g" is normally mentioned as being for "Google", the creator of this framework which was released in 2016. gRPC makes use of HTTP/2 in the background to transport the data instead of the traditional HTTP/1 or HTTP/1.1, enabling for example bidirectional communication and data framing. It also makes use of Protocol Buffers (Protobuf) for data serialization, being language-agnostic and binary-based.

  While gRPC can have many use cases, one of the most seen, like with RPC, is for communication between MSs or between programs in the back-end, in terms of front-end it can face problems in support and compatibility with browsers. An advantageous point of gRPC is that, unlike HTTP libraries, Google provides the necessary tools to use gRPC across different platforms and programming languages, which means a centralized and secured provider of support and improvements, and for having HTTP/2 working on the background, is not impossible to expect a future upgrade to use HTTP/3 announced in ends of 2022, without the need of code manufacturing.

- **WebSockets:** WS is a computer communication protocol created in 2008 and on contrary of the traditional HTTP used by GraphQL and REST API which are Request-Response based, WS is bidirectional, and by opening a single Transmission Control Protocol (TCP) connection which will be kept alive till one of the parts close. It its possible to transfer data in real-time, on a bidirectional basis, and without the need of opening new connections to receive data, it also become the new standard method used for real-time experiences like Web Chats, being abandoned the old ways like HTTP Pooling.

  WS comes with the HTTP/1.1, which provides a mechanism to upgrade a normal HTTP/1

connection to WS, and presents a higher level of complexity than the traditional Web API HTTP request which can be lowered by using some libraries that provide implemented mechanisms, abstraction like Socket.io or services like Firebase API which provides real-time features by using WS and Server-Sent Events (SSE) under the hood, however this last comes with some problems like dependency on the Vendor both in improvements as well difficulty to future migrations, limited freedom and unpredictable costs.

- **Server-Sent Events:** When following the mentality of "using the right tool for the job", SSE can be considered an alternative for WS, being proposed in 2004 by Ian Hickson and first implemented in 2006 by Opera, both make use of different HTTP/1.1 mechanisms to keep alive the connection, but while WS enables a bidirectional communication, which means that the Client and Server can communicate between them freely, till certain degree, with SSE only the Server can keep sending messages to the Client. Some other differences are while WS uses the protocol *ws/wss*, SSE still uses the protocol *http/https* and is also able to sent GET variables when establishing the connection being also easier to implement.

  SSE have however some limitations, it is only able to transfer UTF-8 messages, and when used in browsers, only six connections can be opened at the same time per domain/browser, while there is no limit for WS connections. It can also be harder to maintain a SSE connection opened if not being used, on contrary to WS, this however come with attenuance that per norm, SSE connections are automatically reestablished by the client. This technology is optimized for situations like notification systems or real-time data visualization and can be empowered when used with Publish/Subscribe (Pub/Sub) architectures.

- **Pub/Sub:** Is a message service, being asynchronous and easily scalable, in a Pub/Sub service there are two entities, the Publisher and the Subscriber, the first has the ability to asynchronously broadcast messages in a topic/channel to any present listener, which are Subscribers that had subscribe to that specific topic, and there can be N number of Publisher and Subscribers to the same topic, enabling communication One-to-One, One-to-Many and Many-to-Many.

When broadcasted a message, the Publisher does not wait for an answer from the Subscribers and is unaware on how or when his message is processed. Some common uses of Pub/Sub are to support Load Balancing, Data Streaming, Real-time event distribution or Parallel processing.

- **LoRa and LoRaWAN:** LoRa, which stands for Long Range, is a physical wireless transmission technique first introduced in 2009 by Nicolas Sornin and Olivier Seller, being property of Semtech since May 2012, and is known for its ability, speed and reliability in sending small amounts of data across wide distances by sending information across different frequency channels and data rates while encoding the packets in radio waves using chirp pulses.

  Is considered one of the pillars in boosting the adoption of Massive IoT, in part thanks to being built to require low power and being able to operate with battery devices. LoRa is part of the LPWA, Low-Power Wide-Area, a generic term used to describe any network designed to communicate wirelessly while requiring low power on contrary to others like Satellite, Wi-Fi or Bluetooth, LPWA itself is divided in two classifications, cellular and non-cellular, being LoRa classified as non-cellular and considered by ABI Research as the non-cellular LPWA network technology leader by 2026 [96].

  However LoRa have notable competition like Sigfox and NB-IoT each one for different use cases, being LoRa notable for lower-cost device, very long range (high coverage) and battery lifetime as well as for local network deployment and reliable communication when devices move at high speeds [97], other option is NB-Fi, however when compared with the mentioned ones, is still recent and lacking in detailed comparisons [98] and global coverage.

  LoRaWAN on the other hand is a network protocol that works as bridge between LoRa and the Application being first released in January 2015 and approved as a standard in December 2021 for Low Power Wide Area Networking, or LPWAN, by the International Telecommunication Union, and it presents as a Media Access Control layer protocol to control how the devices use the LoRa hardware, being the transmission method or the messages format.

  LoRaWAN divides the devices in three different classes, depending on how much energy

consume, the time which is "awake" and how long is listening for messages, all three classes can be found being used in diverse scenarios such as Disaster Prevention, Smart Places and Monitoring. Its renowned for is long range, free license, indoor penetration and others more technological aspects.

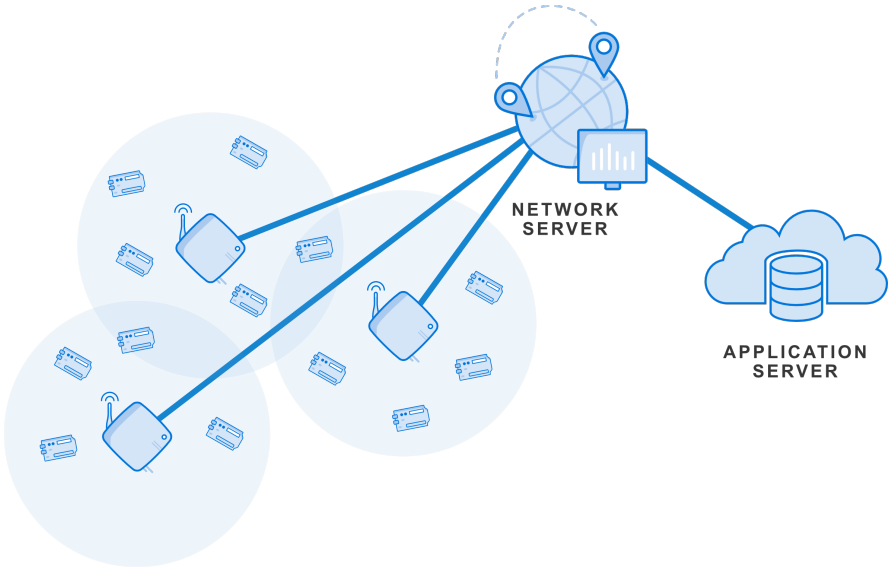

Figure 5.4: Network Architecture with LoRaWAN [99]

This network protocol also makes uses of a two-layer security method by using different keys between the Network and the Application, which means that the Network Server can be unable to read the content of the payload, and just work as a forward server. When using LoRaWAN its possible the use of different protocols to send the data from the Gateways to the Network Server such as Webhooks or Message Queuing Telemetry Transport (MQTT).

The Things Network is a global, open source infrastructure with the goal to provide a open IoT data network, devices and solutions based on LoRaWAN. In a common IoT network is necessary gateways that will forward the data packets from the radio protocols to the Internet, which is possible due to the protocols supporting the Internet Protocol (IP) stack, however LoRaWAN is a non-IP protocol requiring routing and data processing to be received by a application, The Things Network stands as a Network Server in the middle of the gateways, which receives the radio transmissions from the IoT devices or "End Nodes", and the Application Server/End-User applications by providing what they mention as "The

Things Stack", which enables to see and forward the data, as observed in the architecture exposed in the Figure 5.4.

While having as goal the globalization of the IoT Network under a big community network, The Things Network also enables the setup of private IoT Networks under private domains.

5.6 Data Serialization

While the web technologies used are an essential point to take in account, the data serialization will also have a big influence both on the performance as to the difficulty grade of implementation, data serialization consists on the process of converting an object, for example a class object, being its values and properties, into a form that can be stored in files, or in a DB, as well to be transmitted through the network, the most famous used are JSON and Extensible Markup Language (XML). When required, the data can be deserialized in the same program, programming language or another, this enables to transfer data between different services.

- **JavaScript Object Notation:** JSON is a lightweight data-iterchange format created between the years 2000 and 2001 by Douglas Crockfoard as an alternative for XML which was being used since 1998, is a text-based and human-readable way to serialize data to be exchanged between applications and server-client, being supported by most of the used programming languages and the default mode of data serialization of REST API, it is also the format used by Document DB like MongoDB to store the data.

- **Protocol Buffers:** Protobuf was created by Google and released in 2015, being supported across many programming languages, is considered an alternative for JSON and XML to serialize data by offering faster serialization and a smaller payload, this comes by being binary-based instead of text-based, Protobuf also differentiates by only keeping the relevant data, which means that for example the key is ignored on the serialization.

  The method used by Protobuf to enable that both sides are able to understand the data is by using a "contract", schema, which are called Proto files, this files have is own syntax and

by using specific tools for each programming language, code will be generated based on the contract to work with that data, the same files are also used to define gRPC services.

While Protobuf does improve the speed of serialization and data transition while also reducing the payload by almost 70% of how it would be in JSON, it has the comeback of being binary-based which difficult debugging [100], it is also hard to find being used outside of gRPC due to is complexity when compared with JSON/XML.

5.7 Unity Development Tools

Unity is famous in the developers community in part thanks to his ease of use, low learning curve and free plan, being one of the principal choices for small teams or independent developers, while started as a Game Engine for what is known as 2D and 3D games, Unity has being improved and expand its use cases through the development of tools and partnerships with others companies as well, being some of those tools oriented for XR development, being this renowned engine, Unity also tends to be preferable by the XR pioneers.

- **Mixed Reality Toolkit:** Also known as MRTK, is a framework developed by Microsoft released in December 2017, originally, to assist programmers when developing to HL2, its currently on his version 2, being this the version used. While expected to be released the version 3 on February 2023, due to the worldwide situation and Microsoft priorities his development which promised big improvements was delayed to September 2023.

  In meantime, during August 2023, it was announced a partnership with Magic Leap, another leader on MR/AR development, and with Qualcomm Technologies, a renowned name on the XR World, with the goal to turn Mixed Reality Toolkit (MRTK) cross-platform among MR/AR devices. The MRTK provides both scripts, that simplify the use of many features, and UI assets based on the result of their continuous research in UX and UI in MR experiences.

- **XR Interaction Toolkit:** Is an official package developed by Unity which provides the necessary tools to develop for XR applications with a already set of scripts and examples

for most common use cases, this package was firstly released as preview in May 2019 under v0.0.3-preview, being only released a stable version in February 2022, with the version 2.0.0, at the moment the most recent version is 2.5.1 released in September 2023.

This package enables cross-platform development through most of the XR HMD thanks to the integration of OpenXR, however alone, this package is more oriented for VR development, per example, if chosen instead of MRTK3 for HL2 or if for Windows Mixed Reality HMD, it also requires the package Mixed Reality OpenXR Plugin from Microsoft, and if being targeted AR is recommended the use of the package AR Foundation.

- **AR Foundation:** When developing for AR there are some available options, ARCore made by Google targetting both Android and iOS, ARKit made by Apple only for iOS devices and Vuforia, a cross-platform alternative, which on some aspects it may be seen as the better option. However on contrary to the others options, Vuforia is not completely free, being enterprise-driven, and works as a "self" engine, meaning that it uses is own technology to create a AR experience, not being limited to the device hardware, it also provides little support for uses that are not predicted by them.

  Being one of the main reasons for Vuforia to be one of the most used and known tools for AR development the fact that was released in 2011, while the others tools previously mentioned were only released during the years 2017 and 2018 which gave almost six years for Vuforia to establish in the market and set is name, even the antecessor of ARCore known as Project Tango was only released in 2014, other main reason is since they don't depend on the availabe hardware, they support a wider range of devices than the others mentioned SDK.

  If developing in Unity is also possible to use their own package, AR Foundation, this package was in preview since April 2018, when changed the name from ARUtilities to AR Foundation, till February 2019 when released under version 2.0.1. Currently is in the version 6.0.0-pre.3, released in September 2023, being the latest stable release the version 5.0.2 from September 2022.

  While an alternative, AR Foundation does not work standalone as Vuforia, instead is a

"wrapper" of ARCore and ARKit, when developing with this package, at the moment of building the functionalities used will be replaced by their equivalent in ARCore or ARKit depending for which platform is the application being builded, while Vuforia will use the best between the third-party SDK and their own technology.

AR Foundation contain many of the expected features that could be expected for AR development, being listed on their documentation, such as Plane Detection, Anchors and Tracking.

- **Tobii XR:** Is part of the Tobii AB, simply known as Tobii, which is a informatics company focused on the development and research of the Eye Tracking technology, and acts on diverse fields and for various markets, being one of them the XR market.

  When developing for a XR device like the HTC Vive Pro Eye that uses technology from Tobii, its possible to use the Tobii XR SDK to make use of the data acquired thought the Eye Tracking, data like the focus position and which 3D objects are being looked at, between others possibilities presented on their documentation.

  Other option is the use of the SRanipal SDK which provides more raw data, being Tobii XR SDK close to an abstraction, one situation where SRAanipal SDK may be preferable is for example to track the eye focus on a 360º Video.

Aside the tools developed by Unity itself or third-party companies, the Unity Asset Store contains a load of packages that empower the engine by adding new features, adapting external libraries or simplifying Unity components.

- **Best HTTP/2:** While the Unity Engine provides the necessary tools to connect to the network under various protocols, this is a component that suffer big changes under versions and can be hard to do maintenance, among the assets presents on the Unity Asset Store, the "Best HTTP/2" created by Tivadar György Nagy [101] is one of the most known and the best choice when is necessary to connect to the network thanks to his continuous support, simplicity and adaptability across versions and protocols innovation, this asset as per September 2023 costs 59.88 euros.

- **OpenCV for Unity:** While not being one of the expected uses of Unity, the Game Engine has the capacity of making use of CV by capturing frames/photos from the camera, this however can be difficult to implement in C#, the asset "OpenCV for Unity" by EnuxSoftware [102] comes with the necessary code to integrate and adapt the open-source library OpenCV, initially developed by Intel, to be used in Unity, while also providing working examples of different use cases, as per September 2023 this asset can be found at costs 94.81 euros.

5.8 Front-End Web Libraries

The developed prototype web platform has as goal to support the XR applications and enable some features, as such it was chosen technologies that permit a fast development while presenting an acceptable UI for prototypage.

- **React.js:** Like Vue.js and Angular, is a framework created with JavaScript which works by making use of components that work as JavaScript functions and states, which can be comparable as variables, it enables a fast development while providing scalability and ease on maintenance in the long term, the interface is mounted mainly through JavaScript using Client-Side Rendering (CSR) afterwards.

  It also allows to create single-pages web applications without the use of more complex methods like AJAX. Other option putted in balance, which would require learning was the use of Next.js, a framework based on React.js, however Next.js is mainly based in Server-Side Rendering (SSR) and Static Site Generation (SSG), providing more benefits when the goal is to develop websites which benefit from Search Engine Optimization (SEO) and rendering speed like blogs and marketing websites while React.js is better oriented for highly dynamic websites, like social media platforms or dashboards.

- **Material UI:** Also abbreviated to MUI, Material UI is a component library which follows the Google's Material Design rules [103], a set of rules to guide the use of typography, grids, space, scale, color, and imagery when designing interfaces. While not offering big

options of style customization without deconstructing the library, it provides a comprehensive collection of prebuilt components ready for production being a preferable choice for prototyping or scenarios where usability is more important like Backoffice's.

5.9 Back-End Tools and Libraries

This section will highlight some components used in the back-end overall, while being a large component in the project that involves many libraries and tools, only the ones more relevant and that can be of interest to the reader for others use cases will be mentioned.

- **Nginx:** A multi-use tool, it works as Web Server and in this project Nginx is used in the Cloud-Based Server to do Reverse Proxy and Protocol Upgrade to enable and control the multiple WS services developed. While not necessary on a Development state, it was also used Load Balancing in some WS service as to approach a Production state, which may require a different programming approach depending on the kind of WS service.

- **Ngrok:** A tool that enable applications and virtual machines, normally limited to the localhost or the private network, access to the World Wide Web (WWW), being a tool necessary to enable the Web API and the WS services to make use of the MSs that are running in the On-Premise Server.

  Ngrok have a free and a paid version, being the main limitation of the free version the possibility of only opening two network ports to the WWW, by being necessary one for SSH access to development, there was only the possibility of one more network port to enable the consumption of N number of MSs, which was the reason of the development of the MS named "Image Inference" that works as API Gateway.

  Another approach could be the use of Virtual Private Network (VPN), but the free version of Nginx only provides dynamic public IP address, being a constraint in the long term.

- **BoofCV:** A open source library [104] to use CV in real-time and for robotics applications, made from scratch in Java by Peter Abeles for high performance and ease of use, while

offering many features such as image processing routines and image stabilization, the main use in this project is to decode QR Codes and Micro QR Codes.

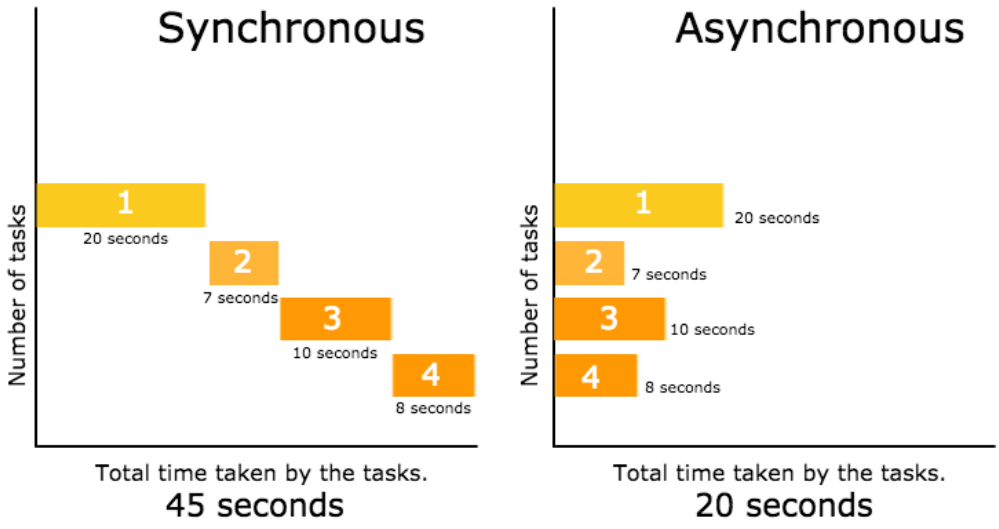

Figure 5.5: Synchronization vs Asynchronous [105]

- **DeepFace:** Made in Python, DeepFace is a lightweight framework created by Sefik Ilkin Serengil that provides the necessary tools for Face Detection, Face Recognition [106] and Face Analysis [107], like age and gender, by wrapping state of the art models, it requires little code to setup a simple application that use this features. However it was done reverse-engineering in this framework to extract only the necessary components, removing unnecessary code to only be used what needed to answer the requirements while also being adapted to use asynchronization (Figure 5.5).

Chapter 6 Prototypes Implementation

This chapter will describe the related prototypes developed based on the architecture and the technologies proposed, the developed components stands as prototypes for being objective in full filling the necessary goal while ignoring minor/standard features, but at the same time also aspiring a production scenery.

The presented conceptual model, aside of XR, also involves a variety of fields that surpass the Author's experience, at the same time the expected scenario is not a contained and limited one, but instead a global and dynamic.

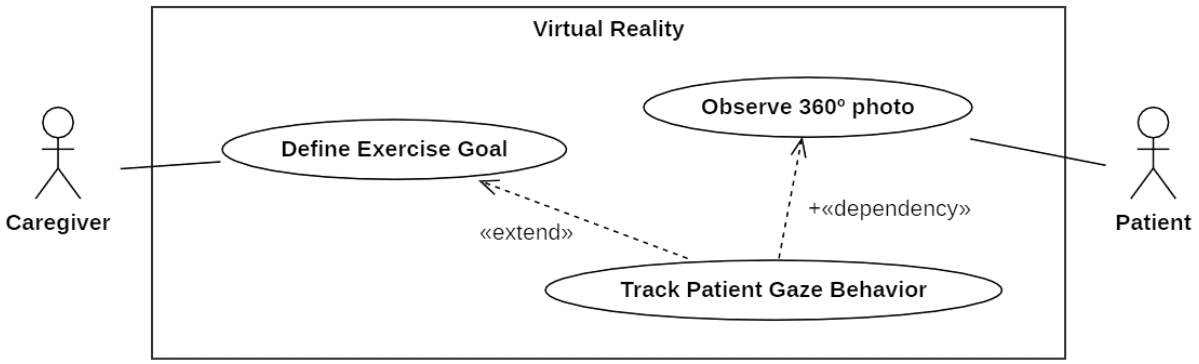

Figure 6.1: Virtual Reality Scenario Use-Case Diagram [Author's Source]

To simply the process of prototyping four uses cases were idealized, use cases these being applied in a scenario of a nursing home, improving elder's well-being and empowering the caregivers. Among those use cases, one involves VR (Figure 6.1) while the others three are with MR (Figure 6.2) being further explained as the follow:

- While supervised by the caregiver, an elder with dementia will be exposed to a series of 360º photographs (involving family, childhood places, ...) through the use of a VR HMD, during that time the caregiver will make questions to the patient with the goal of analysing his current mental state and ability in identifying people, for example;

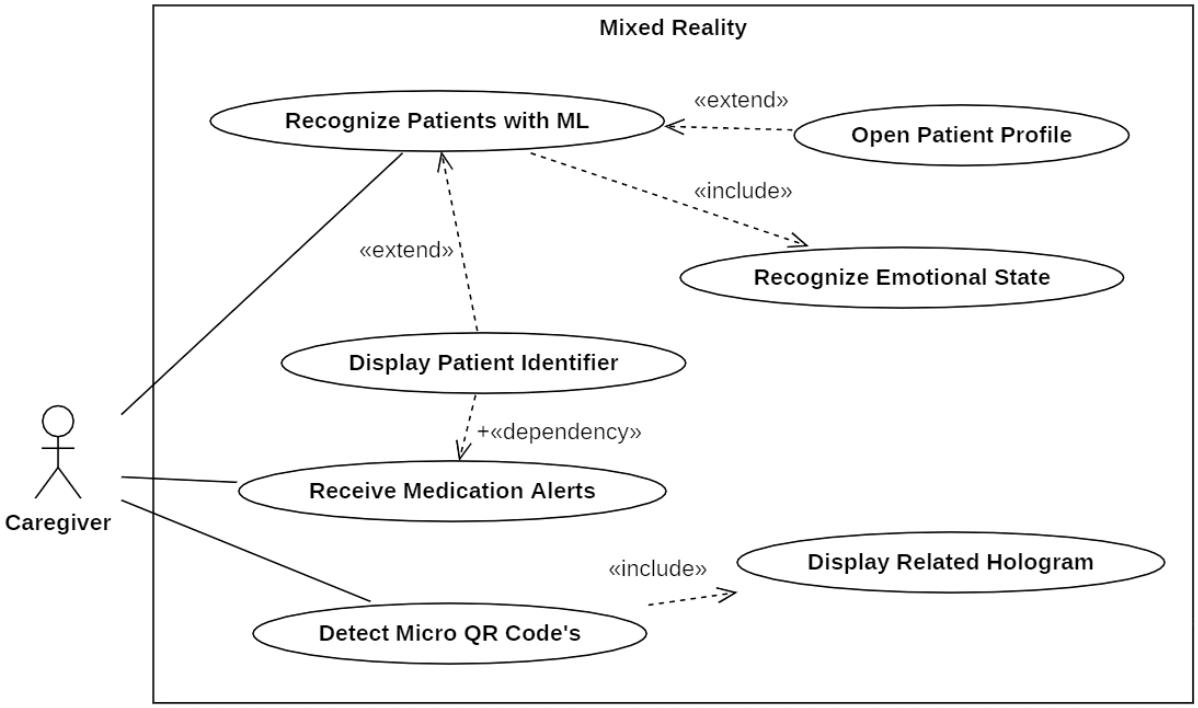

Figure 6.2: Mixed Reality Scenario Use-Case Diagram [Author's Source]

- During the daily work, the caregiver will use a MR HMD being able to look to his patients and recognise his emotional state with ML, he will also be able to access the patient data through the use of Face Recognition and holograms;

- While using the MR HMD the caregiver will receive notifications, alerts such as related to medication to give;

- The elders rooms in the nursing home will be popularized with sensors able to detect temperature and humidity, at the same time in the outside of the rooms, there will be a Micro QR Code which the caregiver can read through the MR Application, being displayed the sensors data through a hologram;

6.1 Identified Obstructions and Constraints

- **UWP Build, WinRT and IL2CPP:** When developing on game engines like Unity, testing can be done within the engine itself. This also applies to developing for HL2, which offers an additional option to test using the `Holographic Remoting Player` application provided by Microsoft.

  By installing this application in the HL2 and connecting both the device and the computer running Unity to the same network, the HL2 can run the application without the need for building. While this option has benefits, there are limitations, such as the inability to use the HL2 holographic keyboard.

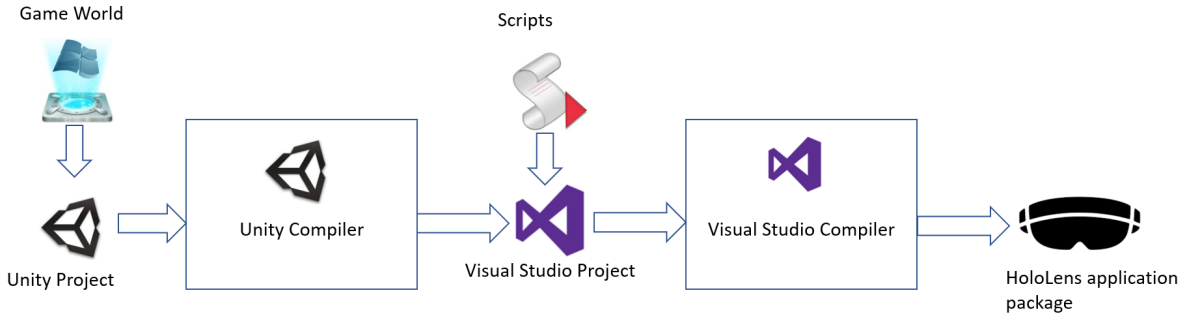

Figure 6.3: HoloLens 2 Build Pipeline [108]

Another limitation is in using the Windows Runtime (WinRT) API from Microsoft which is limited to UWP applications. In Unity, code related to this API must be enclosed within a directive "#IF ENABLE_WINMD_SUPPORT"/"#ENDIF" and will only be recognised during the application build process. This means that neither Unity nor Visual Studio will provide IntelliSense support, and only syntax errors will be identified during the build process. Semantic and logic errors will need to be addressed using a debugger created by the developer, as the built application does not include a console.

While in most use-cases when developing for HL2 this would not prove a problem due to the fact of being rarely needed, this project makes uses of the class `MediaCapture` from the WinRT to capture the frames, instead of more commonly used method, being it the use of

the Unity Engine class `PhotoCapture` which has a bigger impact on the performance or the use of Research Mode which provides more risk and another layer of difficulty.

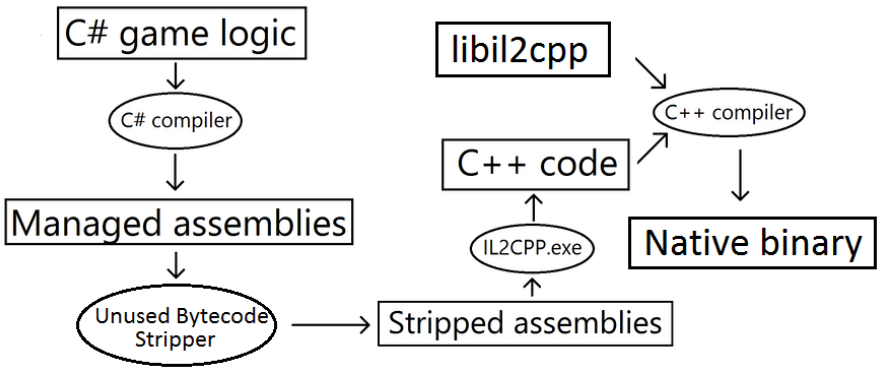

Figure 6.4: IL2CPP Overview [109]

The application build process also differentiates from a normal game/application made in Unity due to the fact that involves two steps of building as shown in Figure 6.3, one in Unity and afterwards in the Visual Studio, in which the Visual Studio Compiler makes uses of Intermediate Language To C++ (IL2CPP) to convert the application to C++ (Figure 6.4). Being this process necessary for any kind of application to be able to run on HL2, all this process, being necessary to test and use the `MediaCapture` class and others features from the WinRT can become time consuming, from the Unity build till the IL2CPP can be used between 3 till 18 minutes being the average time 7 minutes, depending if there is already previous packages build and the changes done since the last build.

- **Augmented Reality Application Compatibility:** While incompatibility with older phones was expected, the same was not with more recent ones, the available smartphone for continuous development and experimentation was a Huaweii P30 Lile, and as it can be seen in the ARCore supported devices website [110] only the P30 or the P30 Pro supports ARCore and with some restrictions, while Vuforia worked, it was alerted by them that the quality would no be good and as mentioned Vuforia is not free and the necessary components, the ability to read Barcodes is part of the Premium Plan, due to this limitation the AR application was limited to conceptualization.

- **HTC Vive Pro Eye and SR_Runtime Disconnection:** In general the HTC Vive products can be considered more Enterprise oriented than Consumer oriented like with Meta/Oculus, however only a few are indeed described as focused for Enterprise, among them the HTC Vive Pro Eye, this come together with known problems which still lacks a definitive resolution, among them is the disconnection of the SRanipal, a software required to use the built-in eye tracking and standalone face tracking module by Vive.

  When read the forums its detected that this problem has being noticed since 2021 and the best solutions are only temporary amends, like starting the SR_Runtime before even the SteamVR, or restart it when disconnected. Others suggestions were the use of older version of SRanipal while disabling auto-update, this however can be challenge for first timers, since the software auto-updates when opened, requiring the modification of the configuration files to disable both software and firmware update before opening, and in case being to late, the uninstalling may not be easy without the use of third-party, requiring the use of the Registry Editor [111].

  The present setup makes use of the latest version of Tobii XR SDK (v3.0.1), with an old version of the SRanipal SDK (v1.3.1) used on previous projects, and while not secure if needed, since the documentation from that time is no more online and the recent one don't mention it, it was also added the package ViveSR, which used to contain plugins related to SRanipal.

  Unfortunately no definitive solution can be expected, while released in 2019, in 2023 this device can already be considered discontinued since its no more found in the official shop of the HTC Vive, and while found on some locations like Asia, its shown as "Out Of Stock" or only hardware details. This discontinuation came in part due to the shortage of Tobii Original Equipment Manufacturer (OEM) parts for the eye tracking, and while it was tried a partenership with a Chinese Tech Company when started the production of the HTC Vive Pro 2, it was a failure.

  Another possible reason may be that while the HTC Vive was the first VR company integrating eye tracking technology in VR HMD, the low number of applications using it,

primarily in entertainment, which is the main market using VR, does not justify the device price and the manufacturing cost, however while the HTC Vive Pro 2 does not come with eye tracking, the most recent standalone VR HMD, VIVE Focus 3 from 2021 and the XR wired/standalone HMD, VIVE XR Elite from 2023, have Eye Tracking.

- **Unity Barracuda and ONNX Models:** After a first implementation it started a process of improvement which ended as is related on this thesis, one of the attempts of improvement was to execute some of the ML services in the XR devices, which was the case of Face Detection by using the WinRT API.

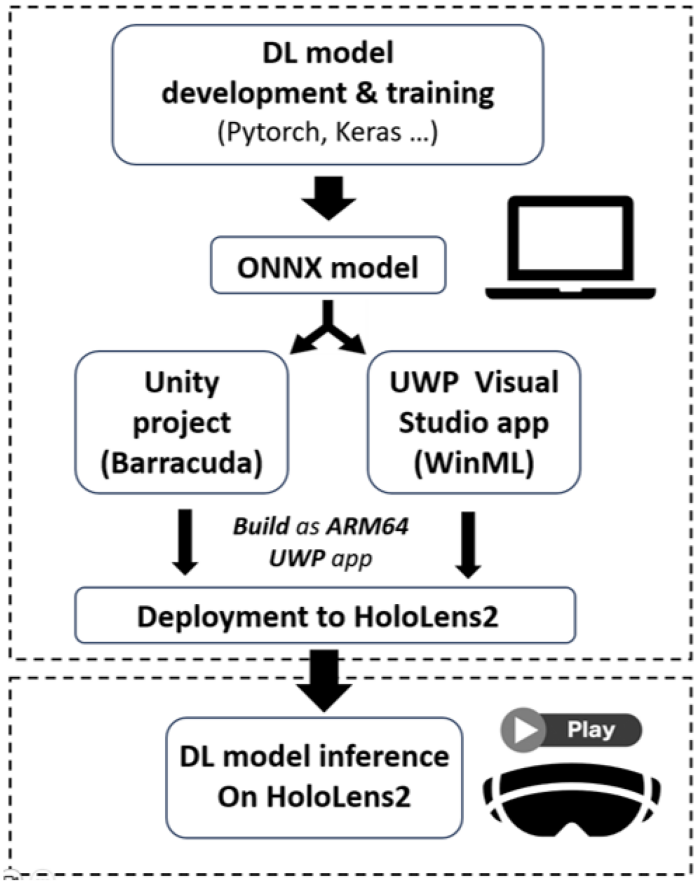

Figure 6.5: Overview of the pipeline for integrating DL models on HoloLens 2 [112]

Another experiment was the use of Emotion Recognition in the HL2 with the Unity lightweight package, Unity Barracuda, a cross-platform neural network inference library that can run neural networks on both the Graphics Processing Unit (GPU) and CPU by using Open

Neural Network Exchange (ONNX) models (Figure 6.5), in which the initial goal was to consume models produced by Unity Machine Learning Agents Toolkit (ML-Agents), and being the best choice among the possible ones both in performance and ease of use [113].

However, while Unity Barracuda is theoretically able to consume all kind of ONNX models, due to limitations on neural architectures and models supported (in the current version 3.0.0) it was not possible to create one for Emotion Recognition, but must be noted that were founded models like for Image Classification or Object Detection which worked.

- **Holographic Remoting Player and SRanipal conflict:** This is a specific problem that occurs when tried the use of `Holographic Remoting Player` while having an installation of SRanipal, which is required for HTC Vive Pro Eye, being asked to execute it also. When denied, the Holographic Remoting Player will also not execute, blocking Unity, while if accepted it will execute but with problems in the rendering. To solve this problem it should be added to the MR project the ViveSR package and activate the SRanipal plugin to load on startup.

6.2 Cloud Databases

Since the goal is the prototypage of certain components of the conceptual model it was defined that using cloud services for the DB would simplify the process instead of setting up the DB on a dedicated server, which also lower the requirements of hardware for the project development. Cloud Databases can be presented in two forms, by being deployed on cloud-based servers, like in a VM, which gives a certain more degree of control and responsibility in terms of software, or as service, being mentioned as Database-as-a-Service (DBaaS).

- **Neo4j:** Among the Graph DB providers, Neo4j, an open-source graph database based on Java, stands as one of the principals providers and technology boosters, having as equal focus the development of the DB as well his potential for DS throught the library Graph Data Science (GDS), in this project the Neo4j is considered as the primary database due his potential to store heterogeneous and biological data [114], when compared with the more

traditional SQL DB like MySQL or relationship-less DB like MongoDB, as well its DS and Data Analysis [88] potential.

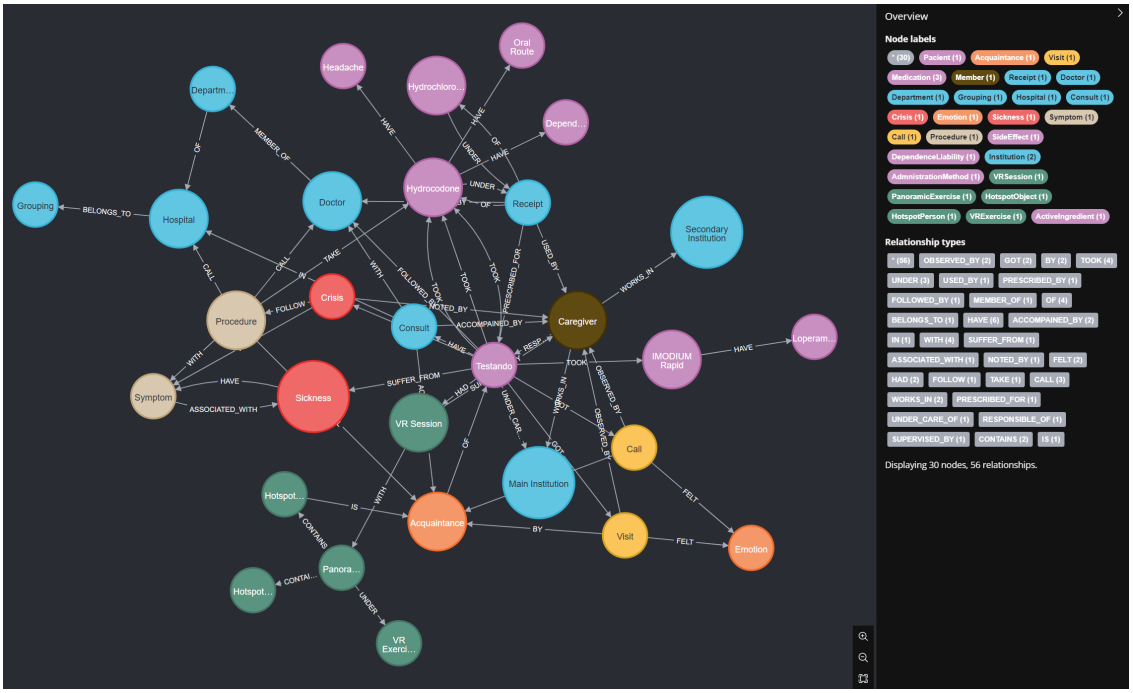

Figure 6.6: Graph Database Design [Author's Source]

The designed database (Figure 6.6) has over 30 entities and 56 types of relationships, however this is but an abstract design, a more in depth would have over hundreds of each, among the entities and relationships designed, the main used were the ones related to person identification and medication related, due to the developed prototypes.

In terms of potential use of DS and Data Analysis its possible to use graph-based path similarity algorithms to create collaborative filtering recommendation systems with knowledge graphs which outperforms both in performance as well accuracy than traditional collaborative filtering algorithm when it comes to commodity similarity accuracy and recommendation result diversity index [115], being possible to create for example, medication recommendation systems that take in account the historic of patients with similar conditions on a interoperable level, Neo4j also can assist in predicting outcomes of decisions taken by using ML [116].

- **MongoDB:** MongoDB is, as previously mentioned, the most famous DB among Document DB and NoSQL DB in general, while able to create relationships between data though the use of the `ObjectID`, this is not recommended for complex systems or performance in terms of scalability, however as with Document DB, MongoDB excels in storing individual data, both in creating and retrieving the data, it also offers a GUI platform named MongoDB Compass.

  This DB was developed to support the Primary DB by storing Big Data or data which does not fit in the Graph DB, the projected collections are related to the developed prototypes, being four collections to store data related to 360º Panoramic Sessions, one stores meta data, like creation date, creator Universally Unique Identifier (UUID), as well data like a list of "to who" is that panoramic session for, as well data related to the bounding box's mapped on the 2D panoramic image.

  The other collection is to store the image itself in binaries, while the common approach is to store the image in the hard-disk and put instead the path in the database, the image is mainly used on devices unable to display images through links, as such is preferential to store the binaries, this is done through the use of the library `GridFS` from MongoDB. By default, a solo document is only able to store till 16 megabytes, and while workable in most cases, panoramic images, even with low quality surpass that limit most of the times, and by being targeted to being used in VR, is required high-quality images.

  `GridFS` works by creating two collections, one collection (COLLECTION_NAME.file) focused in store image metadata, such as total size and total of `chunks`, and the second collection (COLLECTION_NAME.chunks) being to store the `chunks` of the image, each with a limit of 255 kilobytes and storing the `ObjectID` of the related document from the `.file` collection, `GridFS` also works similar to the `sharding process` of MongoDB which results on better data distribution and performance while avoiding a large file system [117].

  The fourth collection stores data related to the bounding box such as content, if applied, type and UUID, this UUID would be used and stored on the Neo4j DB later. While not related directly to the Panoramic Sessions, it was also created a collection to store the logs of the

sessions resulted from the VR application, the stored data would be posteriorly cleaned and stored in the Neo4j DB to use GDS algorithms.

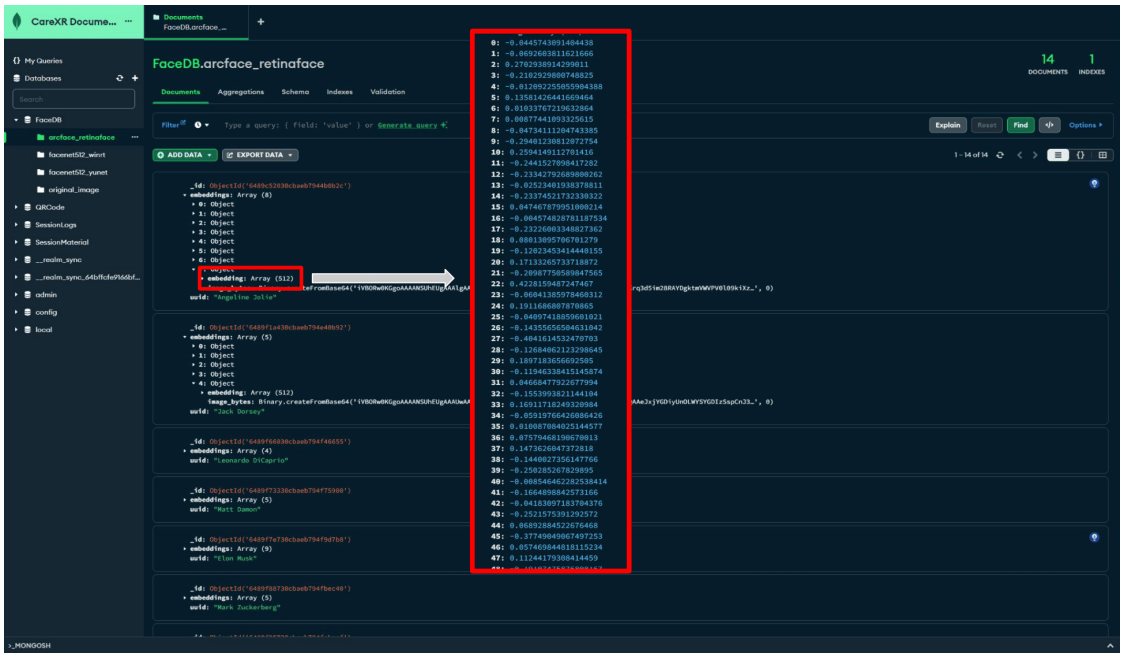

Figure 6.7: Storing Face Recognition Embeddings in MongoDB (Test Data) [Author's Source]

Aside the Panoramic Sessions related data, it was also created collections to store the `embeddings` for Face Recognition, each with a different combination of Face Recognition and Face Detection models, more specifically each document in the collections contains the UUID of the person, being this UUID from the Neo4j DB, and an array of arrays of `embeddings` as well the binaries of the used images (Figure 6.7).

A last collection was created to store data related to (Micro) QR Codes, more specifically the Institution to which belongs, the NanoID and the expected behaviour, functionality, such as display data from sensors retrieved from the Time-Series DB or a panel to control some IoT device.

- **Redis:** Mainly a Key-Value DB, Redis was released in April 2009, however has being expanded to be able to simulate different kinds of DB such as Document DB while keeping the distributed and in-memory concepts, in this project, the cloud version of Redis is being used as Pub/Sub broker to support WS load-balancing. As per GUI platform it was used the

RedisInsight.

- **InfluxDB:** Its a Time-Series DB originally written in Go and first released in 2013, being started in 2019 a refactoring process to Rust, a blazingly fast and memory-efficient programming language, under the version InfluxDB 3.0, having already released the new InfluxDB Cloud Serverless and InfluxDB Cloud Dedicated in early 2023. While others products of InfluxDB 3.0 also rewritten in Rust are to be released during 2023 and 2024. Aside for being the preferable DB provider among others Time-Series DB providers, InfluxDB also shows better performance than others DB highly used in IoT systems like MongoDB and Apache Cassandra when it comes to handle time indexed data [118].

6.3 Servers

While in the firsts stages of the project it was used a VM with Debian 11 as on-premise server, some limitations appeared that made necessary a different approach, one of them being the network security measures implemented in the workplace network which unable the communication between a Web API and an application running in the HL2 even if under the same network. As such the server components were emigrated to a cloud solution, under the available options such as Oracle Cloud or Google Cloud it was chosen the AWS since it was more acknowledged among the searched forums, however the whole back-end component already surpassed the Free Tier limits in terms of size, as such it was necessary a refactoring of the back-end which lead to a necessity to search about different and technique's ending as described in 6.4.

In terms of servers it was used two servers (Figure 6.8), a on-premise server with Debian 11, which resumes in a VM running under VMware Workstation Pro with 4 processors (2 processors and 2 cores per processors), 8 gigabytes of RAM and 128 gigabytes of storage memory, this latest needed a expand during the project implementation. The host computer uses a Intel Core 17-10700KF CPU @ 3.80Ghz with 8 cores and 16 logical processors, while being able to dedicate more resources to the VM this was also the computer used to develop in Unity which also requires a certain degree of resources.

The on-premise server was used to run the largest and heavier components of the back-end,

being them the ML components, these were divided in MLs (6.4.2), and when implemented others MSs those were also kept in that server.

Under normal scenarios a VM running on a computer is unable to be exposed to the Internet without some configurations in the network such as Port-Forwarding and a Bridged Network Connection, this configurations were not possible to implement as such it was searched others possibilities being suggested by a colleague of the Master Degree the use of Ngrok which creates a "tunnel" to the Internet (Figure 6.9).

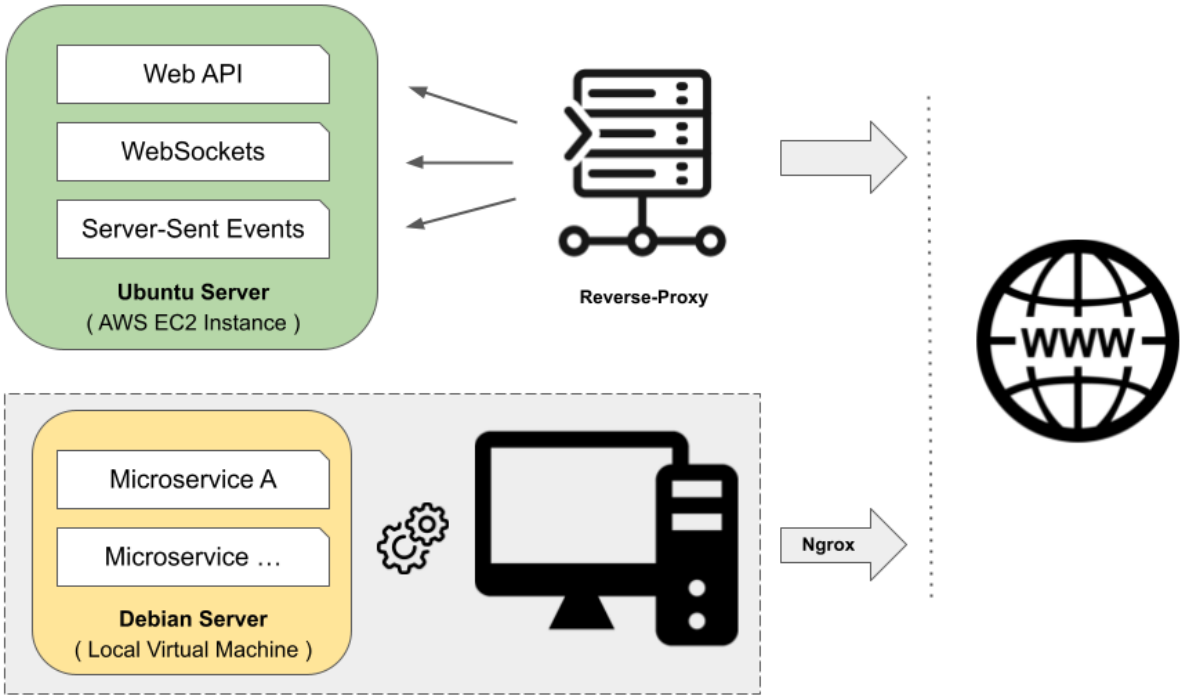

Figure 6.8: Cloud-Based Server and On-Premise Server [Author's Source]

The cloud-based server is a EC2 Instance, more specifically a type t2.micro, which has 1 CPU, 1 gigabyte of RAM and 30 gigabytes of storage memory, being this the limit of the Free Tier, this instance runs the lightweight components such as the Web API and WS, it uses Nginx to provide the Web Server and to use Reverse-Proxy and Load Balance.

While AWS was used during almost all project, the Free Tier has a limited duration, by the end of it, it was tried a migration to the Oracle Cloud, since it was seen as the second best available option and also had the advantage of not having limit time, however the process of creating an

account is very complex and restricted to avoid abuse and it was not possible, as such the components of the cloud-based server were migrated to the on-premise server while still keeping the same flow.

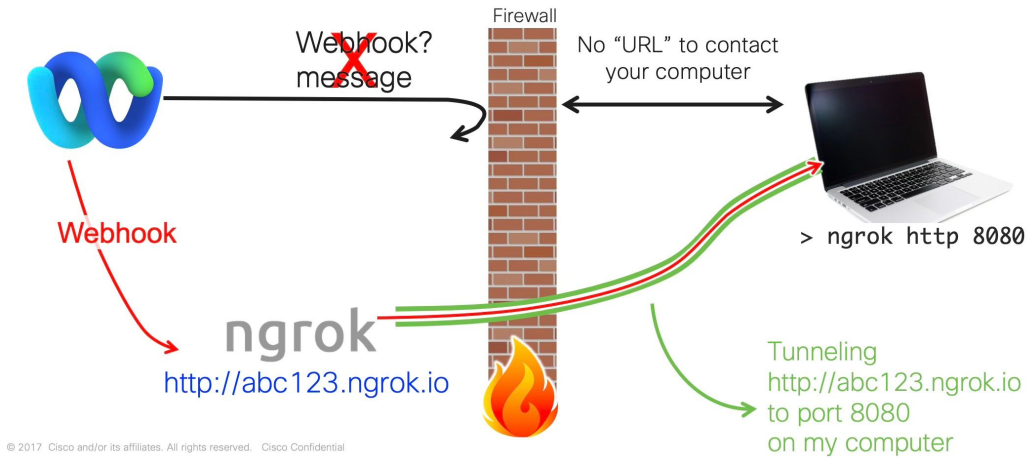

Figure 6.9: Ngrok Tunneling [119]

6.4 Back-End Services

The Web API was subject to some drastic modifications during the project, as it can be seen in the Table 5.1, the first Web API developed was in Python using the framework FastAPI, from Sebastián Ramírez, which is considered the easier and faster to develop with, when compared with others more known Python frameworks, like Flask and Django, while also providing asynchronization by being used with Uvicorn, an Asynchronous Server Gateway Interface (ASGI) web server.

This Web API, already made with GraphQL, aside from connecting with the Neo4j DB and MongoDB DB to do some basic CRUD operations, had also already features related with the consumption of ML models and WS being developed under a monolithic architecture.

When the refactoring of the Web API, it was taken the opportunity to test with ASP.NET Core, due to most of the project being in C#, and Go, which was a well mentioned language, when tested with GraphQL and a basic Login route, which tends to take some time due to the use of Bcrypt for

encryption, Go was faster, and together with the favorable size of the API it was decided to keep with Go for the main Web API, taking the chance to be introduced to Go.

6.4.1 GraphQL Web API

When developing a Web API in Go, on contrary with other languages like Python or C#, its mostly favored by the developers community the use of the native library/package provided by the language, *net/http*, however there are some options like Fasthttp, Gin or Fiber.

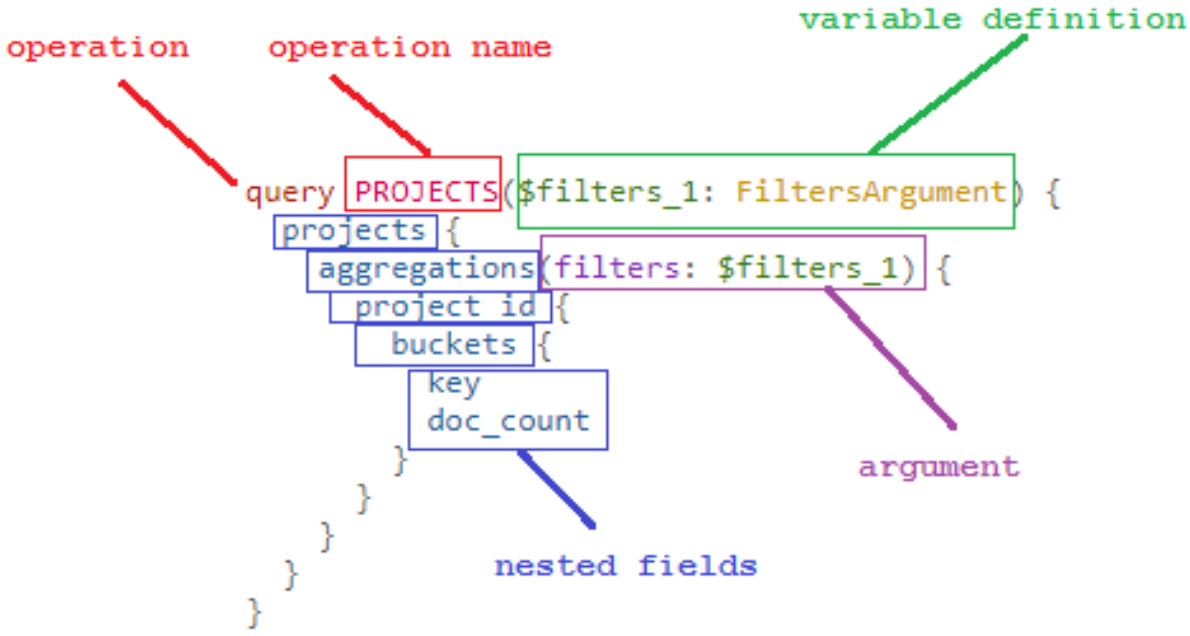

Figure 6.10: GraphQL Query Structure [120]

Being the GraphQL the main requisite it was decided the use of Gin, since it was the only one suggested by the chosen library for using GraphQL in Go, `99designs/gqlgen`. This GraphQL library differs from the others available by being schema-based (Appendix D.4) which means that the developer only needs to write the schema and the library is able to interpret it and generate the skeleton of the necessary code, while updating changes and keeping the unchanged parts of the code, however it does use by default a monolithic architecture, this same schema is also the one needed by others developers to know how to request the necessary data from the API like as shown in Figure 6.10.

Only few methods were developed as was necessary being them like for example, login, JSON Web Token (JWT) creation and validation, get list of available panoramic session and get list of medication to take.

6.4.2 Microservices

When developing applications there are different architecture approaches, which applies also to back-end systems, such as the already mentioned monolithic one, but also others like Service-Oriented Architecture (SOA) which has an enterprise scope or MSs, a subset of SOA, which has an application scope, both have as concept the division of features, however while SOA divides by coarse-grained services like Product Catalog and Shopping Cart features, MSs divides in fine-grained ending with features like Display Products or Check Inventory as individual loosely coupled services.

The three architectures have advantages but also disadvantages, a particular advantage of SOA and MSs is the heterogeneous interoperability, which resumes in the possibility of using different programming languages to better answer the requirements, its also easier to scale and maintain.

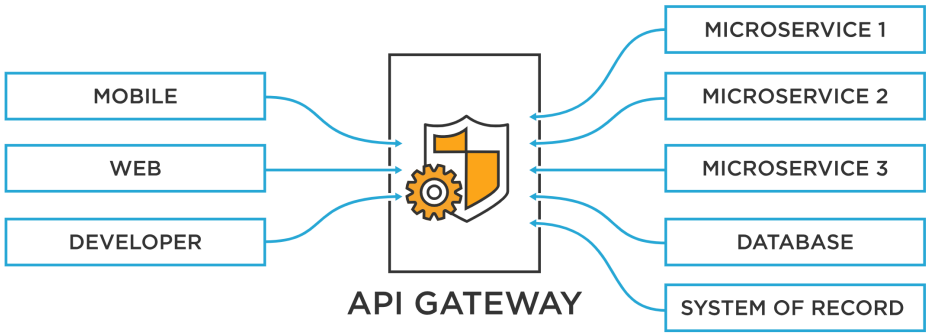

Figure 6.11: API Gateway [121]

The described components are defined as part of a MS architecture instead of SOA since each component works as a single unit and can be used between them, instead of having a Enterprise Service Bus (EBS) or messaging middleware between the consumer and the MS, the use of MSs also simplify the process of debugging or in doing code changes when compared with SOA and also as advantage in horizontal scaling however is known for the increase of complexity.

While in this project MSs are being consumed by others back-end services, its not unusual for them being directly consumed by end-user applications such as websites or mobile applications, which would be the case of some developed MSs if Unity fully supported gRPC as client, this however highlights some of the complexities of this architecture such as route management, security issues, authentication and authorization validation, in those situations is developed a component called API Gateway (Figure 6.11), which stands between the application and the MSs, which means that all requests are made to a specific API which can do the necessary common procedures like authorization or validation and then forward the request for the MSs, it also improve security and performance.

- **Image Inference (API Gateway):** This is a very basic service that works as API Gateway for the MSs, however it was initially made solely due to the limitations in the free version of Ngrok and gRPC support state in Unity.

  To solve the problem with Ngrok it was tried the use of Nginx to do Reverse-Proxies but due to gRPC being basically TCP connections it was not possible, the use of VPN was also a possibility, but in the long term would become time consuming since the Ngrok provides dynamic routes, and the server would be rebooted many times.

- **Face Recognition:** Its a gRPC MS in Python which makes uses of the open-source framework DeepFace, this was however refactored, being extracted only the necessary code from the original version and adapted for asynchronous programming to optimize the code.

  This MS was heavely changed during the evolution of the project, in the start it was also responsible of Face Detection and the Face Recognition was made with the embeddings saved in the disk as in the samples, however after some tests with different models to do Face Detection such as HaaR Cascade from OpenCV and RetinaFace, which resulted on unsatisfied results, the Face Detection was converted on a solo MS, while still keeping methods using the previous models for others use cases.

  The Face Recognition also was changed to be done with the embeddings saved on a cloud DB, more specifically the Document DB MongoDB, for that is used the method Aggregate

from the Python library PyMongo, being the Cosine Similariy formula (Figure 6.12) converted into a DB Query, and being provided the embeddings from the face to be recognized, in the database different arrays of embeddings are grouped in documents, each document corresponding to a person which the only real identification is an UUID of an Entity in the Primary Database, this data is only returned if the document contains at least three arrays of embeddings with similarity greater or equal to 0.85, being the range between 0 and 1.

$$\mathrm{CS}(\mathbf{A},\mathbf{B}) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\cdot\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\cdot\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 6.12: Cosine Similarity Formula

However, due to the lack of comfort of the author in Math and beginner level in MongoDB, as well the lack of online resources, it was used the AI-powered language model ChatGPT to come up with this query (Appedinx D.2), this resulted on a interesting situation since when asked about, ChatGPT affirmed that was not possible to do such thing, only after provided with a example of a basic query using Euclidean Distance made by the framework creator and some dozens of interactions where errors and mistakes were highlighted it was possible to mount the query.

When it comes to extracting the embeddings, three models were experimented, DeepFace by Facebook, FaceNet by Google and ArcFace, being decided to keep with FaceNet model.

- **Person Detection:** This service is a gRPC MS in Python wich makes uses of the ML model You Only Look Once (YOLO) v7 [122], the more recent at the data of development, trained with the COCO dataset, its a very basic MS that receives a bytearray of a image and returns an array of bounding box's, being possible to filtrate persons, ignore persons that are "inside" televisions or laptops.

- **QR Code Decode:** A gRPC MS in Java which makes use of the library BoofCV to decode QR Codes, this library, together with gRPC and Protobuf, is acquired through Maven, instead of the default Gradle, and while possible to decode both QR Code, as Micro QR Code

in the same service, it was decided to separate things.

The creation of a MS for QR Code decode brings some positive points such as cross-platform support, by not depending on a Microsoft plugin (6.5.2). At the same time, after some tests with a QR Code and Micro QR Code with the same content and measuring 8x8cm, it was noticed that the Microsoft plugin were able to detect when closer than 1m of the code, while the BoofCV was able within more than twice the distance.

- **Face Detection with YuNet:** The Face Detection as component of the project also evolved during the project, before being treated as individual component, by depending on how the feature that requires this was being developed in the MR application.

  During the first stages, when the tracking in the MR application was with the use of OpenCV for Unity asset, this component was a gRPC MS in Python that received an image, and would use the cascade-classifier, HaaR, Local Binary Pattern (LPB) or the models RetinaFace, ArcFace, DeepFace or FaceNet to retrieve bounding box's in runtime, being them used to extract the face embeddings afterwards, however the implemented methodology in Unity was not supported by the HL2 due to overheating and hardware capacity, which made necessary a different approach. The final solution used in the application makes use of the WinRT class `FaceDetector`, and aside of the image it also sends the resulted bounding box's from that class, however `FaceDetector` is unique to UWP applications and the used model by this class is not for public access being also the one used by Azure Services, which means, cannot be used in other environments, and UWP till the date also do not support gRPC.

  While not being needed during the runtime since the bounding box's are provided by the Unity itself, it is necessary, recommended, to use the same Face Detection model to retrieve the embeddings to be saved in the DB, as such the back-end would also require the use of the UWP class to populate the DB, however this platform was made to be used in front-end applications, and no good solution was found in terms of back-end, a possible approach would be the development of a front-end UWP application, however the left time to the end of the project did not allowed the learning, and even if done the application, this would limit to the use of HL2.

Since the models found, which used Python, did not show both speed and accuracy to accompany the FaceDetector model, others options were searched, among them it was highlighted YuNet [123], a Convolutional Neural Network (CNN) model in C++. Being expected the development of a gRPC service which would receive an image, and after using the YuNet model, return the bounding box's referenced to the faces detected. This service needed to be done in C++ since the YuNet model was also made in C++, and while being a uncommon approach in ML field, where is mainly used Python, this model takes the best of others Python models, more specifically, it can detect as much as RetinaFace while being fast and hardware consumer as the OpenCV cascade-classifier, LPB.

Figure 6.13: The Seven Universal Facial Expressions of Emotion [124]

- **Emotion Recognition:** This is a gRPC MS in Python with the goal to recognise emotions on a non-controlled environment, by that it means that expects to receive a list of bounding box's, aside of the image containing the persons (Appendix D.6). When it comes to Emotion Recognition, the common approach is the focus in facial expression and in the principal emotions (Figure 6.13), which requires a bounding box from a Face Detection inference for example, being both methods provided by the DeepFace library.

Another method, more uncommon, is the focus on the body expression. This MS provides both, on a face-focused its used the DeepFace library, while for the body-focused uses a CNN model developed by Abhishek Tandon [125] using PyTorch, based on the methodology proposed in the same article where the EMOTIC dataset is presented.

The article [126] proposes the recognition of emotions through the analysis of the person's body as well as the environment that surrounds him, by giving importance to the context, thus resulting in two types of evaluations, Continuous Emotion and Categorical Emotion, the latter making use of a list of 26 emotions while Continuous refers to three emotional dimensions, Valence, Arousal and Dominance, each evaluated from 0 to 10, this also highlights this mode when compared with others that focus in the seven universal facial expressions being them happiness, surprise, contempt, sadness, fear, disgust, and anger.

The developed model makes use of the YOLO v3 model as well a CNN model developed on the Place365 dataset [22] with a focus on the analysis of the environment. This later however have a problem since the EMOTIC dataset is not to be used for commercial purpose, being the goal here the research as well the test of the use of this kind of approach.

6.4.3 WebSockets and Server-Sent Events

After the migration of the "body" of the Web API to Go, among the lacking things were the WS, while possible to develop WS in Go, the suggested library does not have maintenance currently which means no support and prospects of improvements, alongside, by being a new and complex programming language to the author together with a topic also rather new and complex, it did not show as having a positive out coming. Being decided the use of Node.js after some research, due to is performance and ease of use, characteristic of JavaScript [127].

While possible to use tools to simplify the process like `Socket.io`, it was decided to use the more "raw" library, ws. Each WS service was developed separately, while being possible to distinct services on the same WS through the acquisition of the Uniform Resource Locator (URL) path, that way is harder to scale and maintain the code, putting also all WS in risk of going down in case one fails, on the other hand, by separating the services on different WS running on different

ports, its the opposite, while also being possible to optimize with the use of load-balance.

- **QR Code Authentication:** A basic WS inspired in Whatsapp and which the flow is shown in Figure 6.14, it is to be used with XR HMD able to capture photos of the real-world, the User after login in the Dashboard is able to generate a QR Code which contains a JWT with a UUID from the Web API, connecting to a WS in a room with that same UUID, that is, the WS for QR Code Auth.

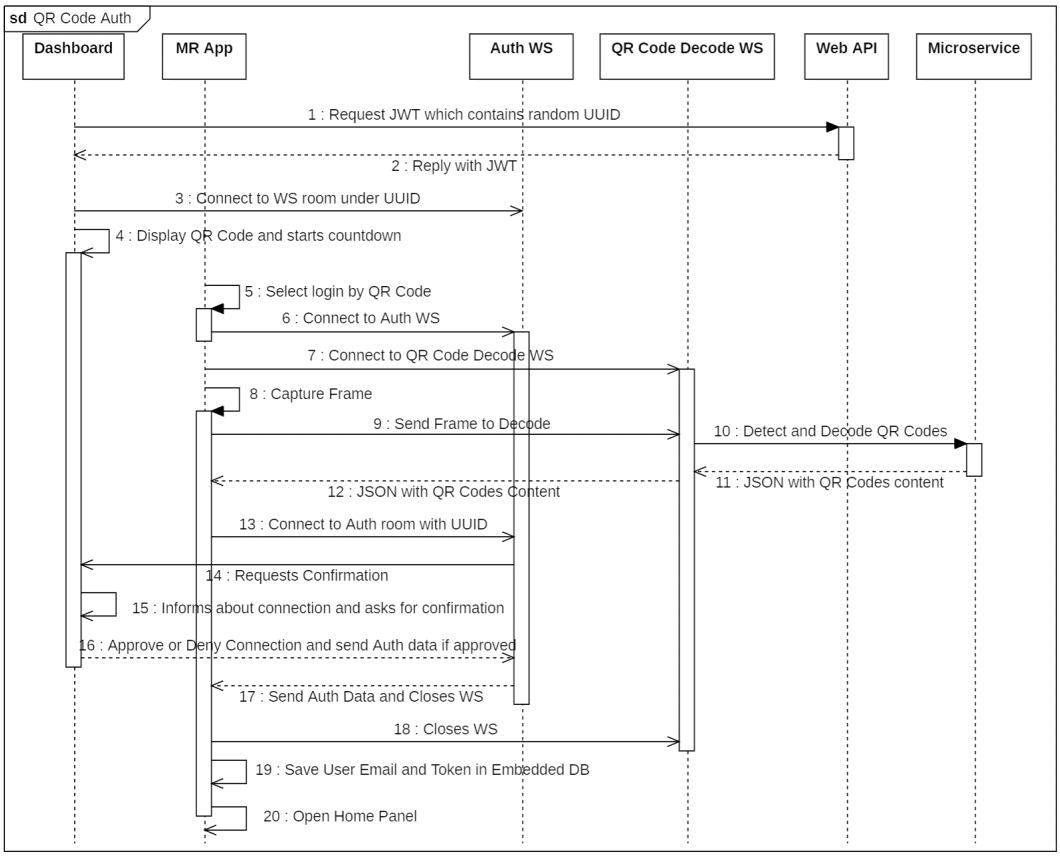

Figure 6.14: QR Code Authentication Flow

Through the XR application camera it will be captured the QR Code which will be sent to a WS and MS to acquire the content, after which, the application will then connect to the QR Code Auth WS and through the UUID to the room channel. Afterwards it asks for confirmation in the Dashboard, which confirmed, will transfer his authorization token to the application closing the WS afterwards. This WS does not make use of Load Balancing however it can be done with Redis.

- **Micro/QR Code Decodification:** This WS has the goal to enable the consumption of the MS of QR Code, its very straightforward, receiving only the image in Base64 and forwarding her to the mentioned MS, replying afterwards with the reply from the MS, the existence of this WS could be avoided in case gRPC was supported by Unity, other approach would be not doing a MS but instead do a WS with some framework in Java, however to simply the process all WS were kept in JavaScript.

- **Face/Emotion Recognition:** Like `Micro/QR Code Decodification` these WS exist to forward the Base64 image to the MS responsible in doing inference of the image for Face and Emotion recognition.

- **VR Panoramic Session Control:** This is the more complex WS developed, introducing the use of Load Balancing and Pub/Sub, it has the goal to enable the User in the Dashboard to control the VR application till certain degree.

  This is done through a schema of a JSON that is sent between the Application, WS and Dashboard, being interpreted by all parts (Listing 6.4.3). The WS is firstly established by the Dashboard being provided a user-friendly NanoID when established the connection, this NanoID is also used to create a temporary channel in the Redis DB at which the Dashboard client subscribes. Being also attributed a UUID to the Dashboard to identify him, after this NanoID being used by the VR application to also connect to the WS its verified in the Redis DB if such channel exists, if not its ignored, if it exists the client Application receives also a UUID as identifier and it subscribes in the Pub/Sub channel.

  After both entities being connected successfully, is upgraded to use a UUID to identify the channel, this process is not seamless, being required that each connection receive that UUID and subscribes to the new channel, during all the process the WS connection pass by four states, Initialize, Upgrading, Connecting and Connected ending in Running which stays in till the WS being closed.

  While before reaching the Running state the WS itself contains some programming logic, most of the logic is done in the Pub/Sub part, more specifically when the WS receives

messages it publishes that same message to Pub/Sub channel, being the component of Subscriber the responsible to filter and process the message. Some of the developed commands are to load scene, download panoramic image and setup hotspots, as well basic commands to pause or save exercise.

```
{  "state": "running",
   "managerUUID": "0c2dcd39-5398-478f-a9fb-689cab91a84b",
   "applicationUUID": "269a98fc-cde1-42a2-ac79-5554620c3b36",
   "execute": {
      "operation": "loadScene",
      "params": { "scene": "Panoramic Session" },
      "requester": "0c2dcd39-5398-478f-a9fb-689cab91a84b",
      "responder": "269a98fc-cde1-42a2-ac79-5554620c3b36"
   },
   "channel": "13d6d308-5a4f-4e38-b8a7-12e4f213790a" }
```

Listing 6.1: Example of a WebSocket Message to execute "Load Scene"

In the case of the exercise `Recognition Ability with Eye Tracking Exercise`, this WS is also responsible for acquire the panoramic image from the MongoDB and sent it to the VR application and the Web Application, while making sure that both applications only start when the image is already loaded in both. In relation to the VR application the image is sent in Protobuf, this was a necessary approach due to the size of the image, however it requires to signal the VR application to be ready to process the Protobuf, the same cant be done in relation to the Web Application so the image is sent in Base64 which would inviable without modifying the image, as such is used the package `jimp@0.22.10` to resize the image.

- **VR Panoramic Session Data Stream:** While `VR Panoramic Session Control` is used to control the session and transfer specific data related to the application state, during the execution of exercises should also be possible to visualize relative data in the Dashboard.

Such data should not be sent through the mentioned WS since it may obstruct the communication, being this WS developed so the VR application can send messages freely since is not expected to receive messages.

This WS also uses Pub/Sub to support Load Balance and the UUID of the channel is shared through `VR Panoramic Session Control`, it shall be noticed that this WS only communicates with the VR application, receiving the data from the application and publishing her in the channel, also at the current state of the project the mentioned data refers to a JSON relative to the execution of the exercise `Recognition Ability with Eye Tracking Exercise` (6.5.1), this is for example, which Bounding Box is being looked at.

In terms of Server-Sent Events it was only necessary the development of one, however furthers features would benefit of this approach instead of WS.

- **VR Panoramic Session Data Receiver:** This is a simple SSE that is connected by the Dashboard, which at the moment of connection pass the streaming channel UUID as GET variable in URL, after which the SSE subscribes to the channel and forwards the messages to the Dashboard.

6.5 End-User Applications

End-User Application is a terminology used to mention the developed application(s) expected to be used by the final user, this is, in a e-commerce platform would be the buyers and sellers, while in this projects are the Healthcare Professionals, Patients and Relatives, being referred in the Conceptual Model four End-User Applications, a Dashboard and three different XR applications, in respect to the XR applications all were developed in Unity and as such some features are shared or reused between them, with the necessary adjustments.

- **Data Persistence with Embedded Database:** When developing for Unity, products tend to be games or serious games, which can require some level of data persistence being the main

methods the use of `PlayerPrefs` or `Save Files`, the first method is ideal for low volume of data and works as a key-value architecture, whoever when necessary to save more complex data the use of `Save Files` are preferential, this can end on encrypted JSON or even Binary, but that still makes the data vulnerable and should be used till certain volume of data.

However when it comes to Android or iOS applications the method used are Embedded DB, which provides more security and ease of use to handle complex and large volumes of data, being the more commonly used the Relational DB SQLite, Unity itself also allows the integration and use of SQLite, but by being a SQL DB it requires to have already an idea of schema, other option is the use of Realm, an Object-Oriented DB released in January 2017, by the same creators of MongoDB, by being NoSQL, DB data structure can easily be modified when necessary, and as others Object-Oriented DB, the integration in the code is quite seamless.

Realm makes use of asynchronization programming when dealing with the data, and while not used in the prototypes, its possible to synchronize the data in real-time with a MongoDB DB. The management of the Realm is done in a static class (Appedinx D.1) and while being mainly used by the MR application, it was setup in the VR.

- **Concurrency and Asynchronization:** When developing in Unity, there is a base class from which all scripts inherit by default, the `MonoBehavior`, among others aspects this class implements two mains methods, `Start` and `Update`, the first being executed at the start of the application, when the `GameObject` is active for the first time in the scene, while the `Update` is consecutively called, all the logic contained in the `Update` method in all active scripts must be executed before being recalled again, this process is part of what defines the Frame Per Second (FPS), also known as Frame Rate, being the term "frame" referred for each execution of the `Update` method. This translates into some limitations when doing implementations, for example, its impossible to do a "fade in" effect or time countdowns, since the implemented `while` or `for each` are expected to be executed the fastest possible, in certain cases the use of this methods can broke the application.

To solve this problem is normally used `Coroutines`, these are pieces of code that are executed apart from the main flow of the game/application enabling defining different kinds of conditions to repeat the code or end it, as well to suspend itself without affecting the main thread execution. However `Coroutines` have some drawbacks, firstly it must be started by a script inheriting from `MonoBehaviour` or from the class `GameObject` itself and while `Coroutines` enable multitasking, it's not multithreading, which means that it does not translates in better use of resources since it runs in the main thread, another problem, while less notable is that `Coroutine` should not take a long time between iterations taking the risk of pausing the main thread, since they are synchronous operations.

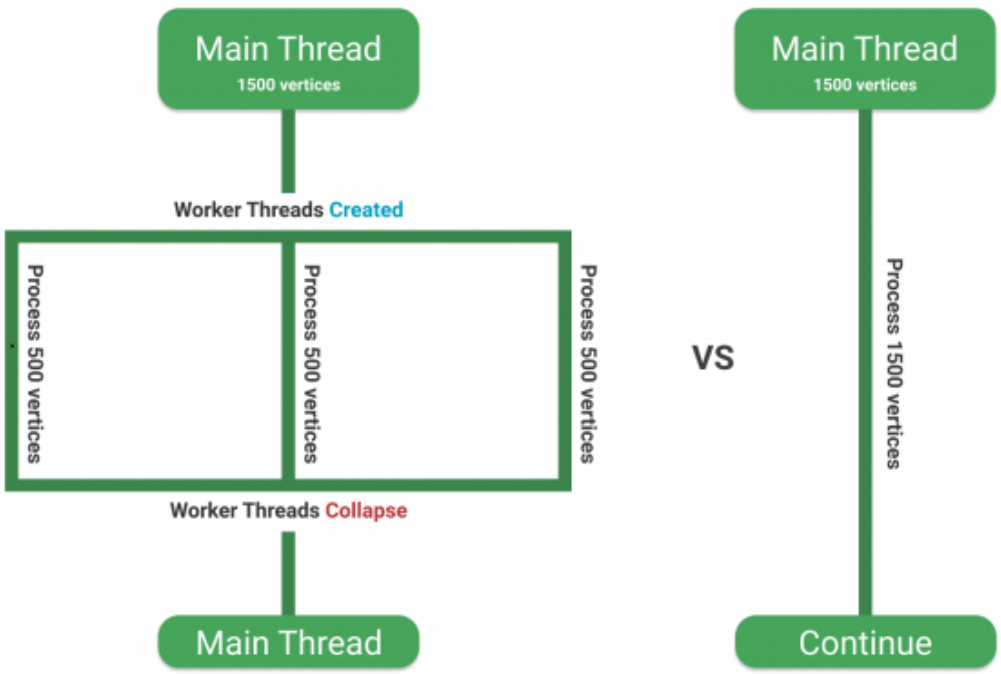

Figure 6.15: Understanding Unity Job System [128]

In relation to the last mentioned problem, a solution would be the use of Asynchronous Functions, which are part of the C# language, these methods make use of the keywords `async` and `await`, being the later used inside of a `async` function to define what the functions need to wait for to continue, such as other asynchronous function or Task, being task part of the Task Parallel Library or TPL, and resumes in an asynchronous method that performs a single operation.

These methods in opposite to `Coroutines` are not limited to run in the main thread and does not have the risk of block it, however threads are created seamless and only if necessary. A last option is the use of Unity Job System, which is the tool provided by Unity to use true multithreading and parallel processing (Figure 6.15), Unity naturally makes use of multithreading but just to some extent while most of the logic is executed on a single thread, the main thread, however the Job System provides support to the developer for easy use of multithreading.

In this project the applications developed in Unity make use of Asynchronization and `Coroutines`, and while it was tried the use of Job System for some heavy methods, it proved harder than expected and as such it was not make use of multithreading.

- **Web Requests Manager:** This refers to a static class that works as a wrapper of the Best HTTP/2, providing some generic methods to handle the management of WS through the use of `Actions` or `anonymous functions`, and Web API requests in GraphQL, in specific it was developed the necessary code to mount the body for GraphQL requests (Appendix D.5).

6.5.1 Cross-Platform Virtual Reality

The developed application has a goal to work as a cross-platform application, by using OpenXR, and to be used in patients to execute exercises or VR activities, this application is made to work as a "portal" to access and run these exercises, activities, which means that those are downloaded or accessed from the web, while the application itself only contains the basic code to execute them. To simplify the process it was used as start the template from Valem, originally made to the VR Jam 2023 [129].

- **Remote Controlled Session:** This, while the most simple functionality, is the principal feature of the application, when started, the user, at that time expected to be a healthcare professional, is putted inside a simple 3D model of a building [130] and is only able to click on a button to create a connection with a WS service, if connected to the Internet (Figure 6.16). This is mostly for standalone HMD while wired one's may just press the "Space" key.

When established the connection, is presented in a screen in the VR a user-friendly code which is to be inserted in the Web Application, this will start the process to establish a WS connection between the VR application and the dashboard in the Web Application. Afterwards, the HMD may be put in the patient being only necessary to retrieve it at the end of the session or in case of emergencies.

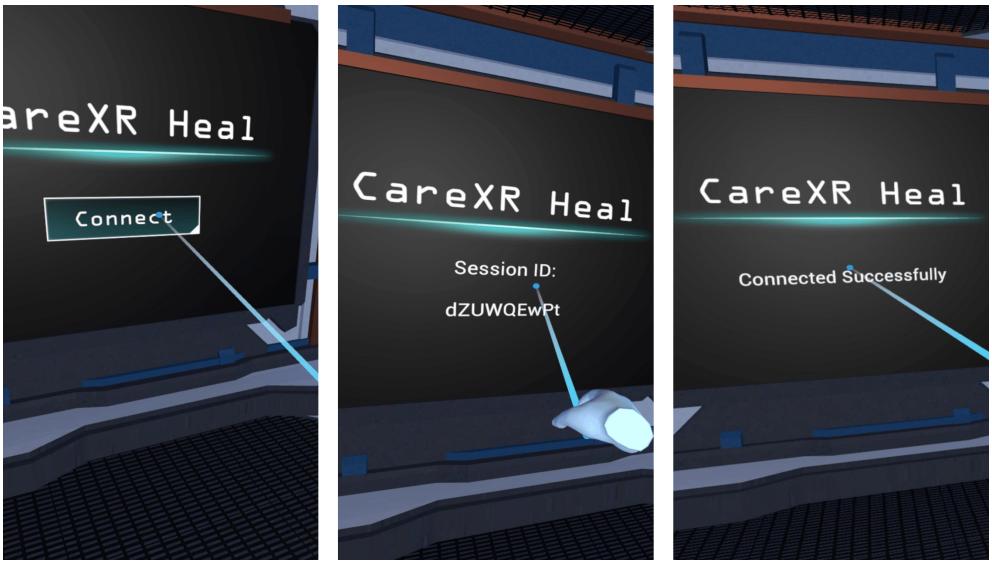

Figure 6.16: Connection Stages in VR Application [Author's Source]

Being possible to execute *N* exercises of different types without the need of removing the device and putting him again, lowering the stress provoked in the patient while improving his acceptance and comfort. The use of this methodology, one application able to run *N* activities, instead of *N* applications, each one with a very specific activity in mind, is to simplify the work of the healthcare stuff, while promoting scalability, it may prove more complex in the beginning when implementing new kind of activities, however afterwards, the same logic may be reused.

When thinking on activities to run on a VR application, two types highlighted, 360º image based and 3D based, in this project the initial goal was to explore 360º image with eye tracking which is explained in depth in 6.5.1, however is also possible to integrate 3D model based exercises, this however would require a different approach, by being the goal having the exercises data in the web, this kind of exercises would require the development of what

is known as Multiplayer Game Servers, by using tools like Fishnet, Netcode or Proton, aside of the Unity application, this may require a new and larger set of WS depending of the type of activity, control expected and data transaction.

- **Recognition Ability with Eye Tracking Exercise:** This exercise makes uses of Eye Tracking to track the user, patient, capacity of recognising X, being persons, objects or buildings in a 360º image. After being selected an panoramic image in the Web Application, this is sent to the VR application through WS in Protobuf. For the prototyping phase, the concept of stereoscopic viewing (Figure 6.17), also known as 3D objects/images, was not implemented, which consists of rendering the same image twice, once for each eye, but with slightly different positions to create a 3D depth sensation, by imitating the way of how humans see.

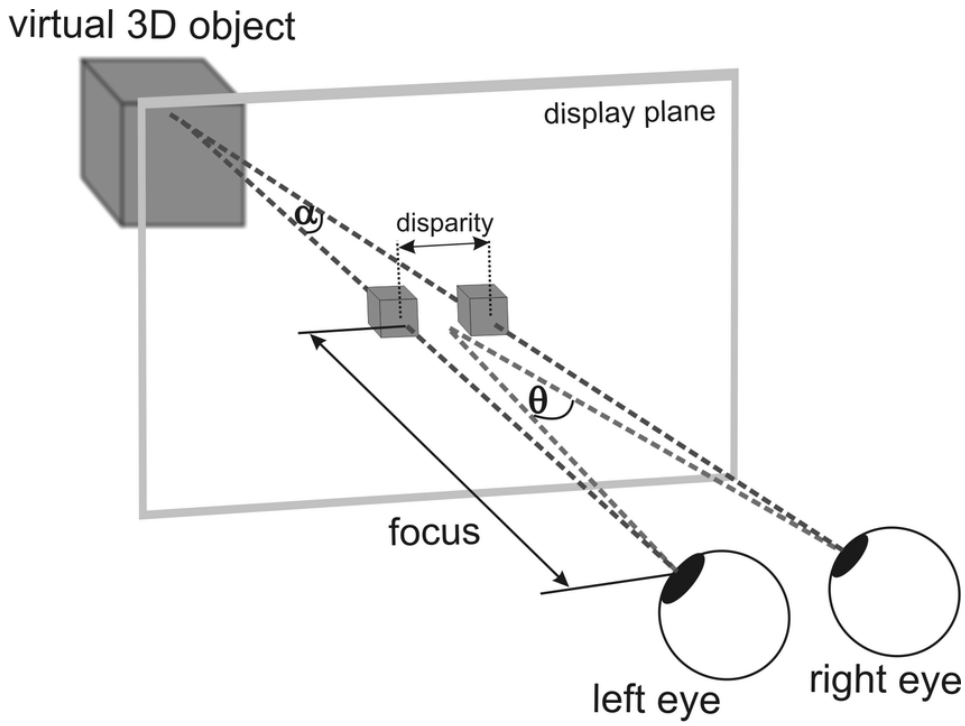

Figure 6.17: Concept of Stereoscopic Viewing [131]

After the data related to the bounding box's of the objects to be taken in account and the exercise are sent in JSON, is initialized the process of building the 360º image, this starts by

transporting the user to a different Scene, specific to Panoramic Exercises, being instantiated a basic `sphere`, the bytearray of the panoramic image is then converted into a `texture`, which is applied to a `material` and finally the `material` is applied to the `sphere`. This however does not work straightforward because by default textures are applied on the surface of the `sphere`, while the required is to fill the inside, being necessary to invert the triangle winding in the code. This process is needed because is being made during runtime, dynamically, if done during the building, statically, it can be used materials as `Skybox`.

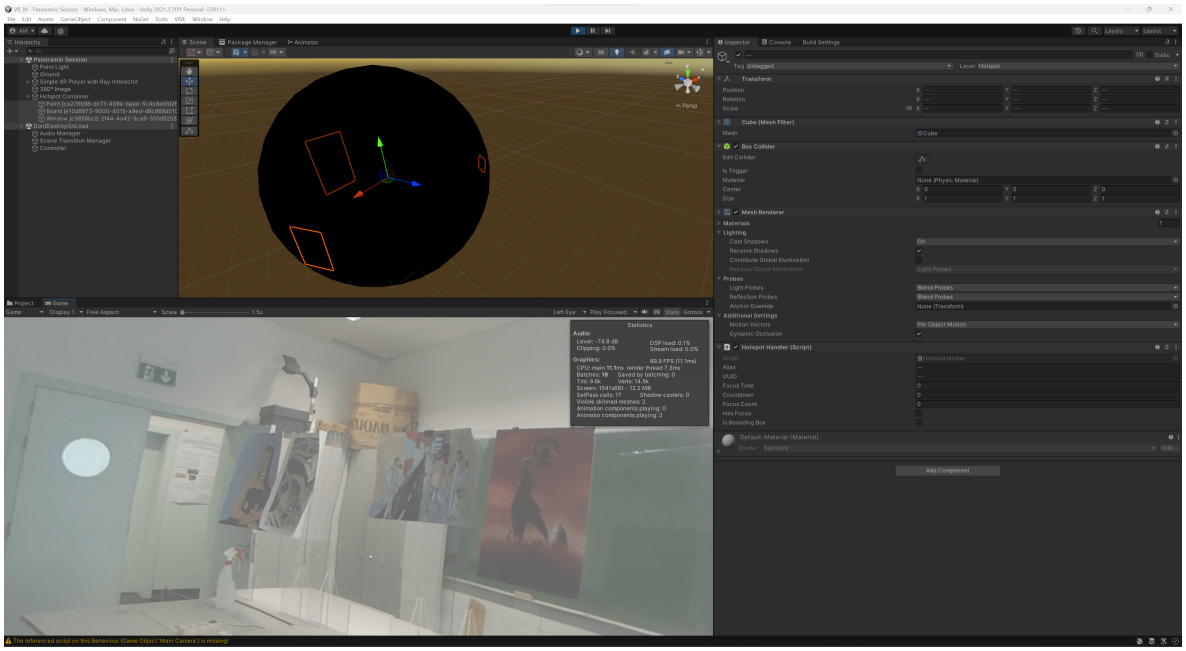

Figure 6.18: Outside View of Sphere and Bounding Box's [Author's Source]

When finalised the `sphere`, starts the process of instantiate basic cubes to work as bounding box's, this starts by calculating the center of the bounding box, being then converted the 2D position to spherical coordinates, and this is then converted to a 3D position in the `sphere` surface by taking in account the radius, this calculations were made with the support of ChatGPT, however he was unable to convert the 2D width and height to Unity measures, then, after different attempts, the same logic was used to calculate the `Vector3`, the 3D position, of the points x1y1, x2y1 and x2y2, being calculated the difference of these point, with the magnitude of the resulted `Vector3` it was possible to scale the bounding box both in width (x1y1 - x2y1) and height (x2y2 - x2y1).

During the creation of these cubes, which are referred as hotspots, is also attributed a component of a class to store the data related to the hotspot such as UUID and label, this class also implements the interface `IGazeFocusable` from the `Tobii XR SDK`, which requires the `GameObject`, in this case the cubes, to have a `Collider` component, which provides a method able to detect if it is being looked at, to improve scalability this method only invokes a dynamic action depending if started to being looked at or stopped, being passed the class (this) as argument. The related code is presented in the Appendix D.7 and the result is shown in Figure 6.18.

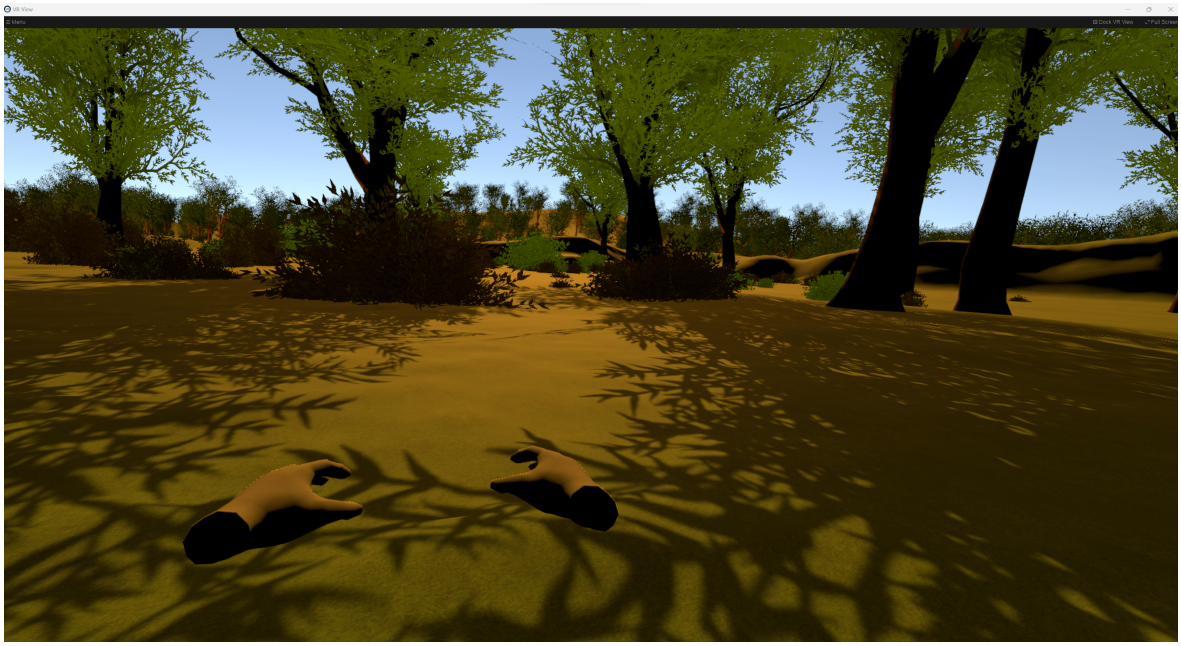

Figure 6.19: Forest in Virtual Reality [Author's Source]

In the meantime, in the Web Application, the healthcare professional is able to chose *N* number of exercises based on the Panoramic image, being the goal the Recognition one, when selected, is started the exercise of recognition. When started the required prefabs from the `Tobii XR SDK` to enable Eye Tracking are instantiated and a new WS is established to stream the data of the exercise, more specifically, which object is the patient looking at, and the action `OnHasFocus` when invoked, starts a `Coroutine` to count the time during which the object was focused and sends to the WS the data related to the object being stared, while the `OnLostFocus` stops the mentioned `Coroutine`, resets the focus time and sends to the

WS a warning that the patient is no more looking to the object.

During the exercise, these interaction are also saved to be putted in the DB at the end of the exercise, if intended. The Unity application is also able to execute some commands during the exercise, such as Pause/Continue, Restart, Save Data/Ignore or Stop, when triggered in the Web Application, these commands are transferred in the session control WS.

In the case of the Stop command, the exercise is taken as complete and the patient is transferred to a different Scene, referred as Lobby, where is surrounded by a virtual [132] forest and can move freely (Figure 6.19), while the healthcare professional can see the exercise log, choose a different exercise or conclude the session, this Lobby is also accessible in the beginning of the session if triggered in the Dashboard.

6.5.2 Mixed Reality with UWP

While the goal of the project was to develop cross-platform applications, the available equipment supporting MR, the HL2, runs with UWP, which provides some native features not present or usable in others HMD like the Meta Quest 3, and since the hardware did not supported some features when using a cross-platform approach it was used a device-based approach. At the same time, as mentioned in 2.1, the idea of MR can be approached by two different ways, which can also make necessary different approaches in development, however without a Passthrough based device it was not possible to test such differences.

UWP is an alternative framework to develop client applications in Windows, instead of WPF which stands for Windows Presentation Foundation or the more known Windows Forms, this alternative stands above the others in terms of UI and when targeted for internet-connected devices, those same applications can also run easily in others Windows 10 devices, UWP was first introduced in 2015 and was the focus of Microsoft when released the HoloLens 1 in 2016 and the HL2 in 2019, however during 2021 rumours about UWP being abandoned started to appear in developers forums and during 2022 this was confirmed, being UWP deprecated in favor of a new framework named WinUI 3 and the desktop-focused Windows App SDK for Windows applications development, previously known as Project Reunion, while the repercussions of that transition

in a future HoloLens 3 are undetermined, it can means a good future for cross-platform development, the same also can be predicted since the partnership with others companies to improve MRTK.

A necessary point to highlight is that the current interface when captured through the HL2 camera or in the Windows Device Portal shows rather low quality (e.g. Figure 6.25), this is mostly due to the use of the color black as background of the interface, requiring a little image edition, that same interface can also be viewed in the Appendix E.8 as it is in the Unity. While the color black can be seen in the Unity Editor, when running the application the color black is not rendered due to limitations of the MRTK2, creating a sensation of transparency, however MRTK3 is able to render a dark grey when used the recommended materials. Aside from colors, it was also noted a discrepancy in some holograms position when comparing the view and the photography.

- **QR Code Reader:** This feature is meant to read QR Codes present in front of the user by using the frames captured through the class `Media Capture` from the WinRT, a different approach would be the use of the class `Photo Capture` native from Unity, however this class is not performance-friendly and it may even pause the application during the moment of capture, this method however is limited to HL2. It also requires careful implementation since only one instance of `Media Capture` can be activate at the same time and the same instance is also used for `Face and Emotion Recognition`, afterwards, the QR Code content would then be processed by the application, the main goal being the creation of specific interactions for the MR user, such as the creation of a panel.

  When it comes to read and acquire the content of a QR Code, the HL2 is able to do that without being required by the user, that is, when in the "Home" of the HL2, he automatically does the recognition, displaying a bounding box with a play icon hover any detected QR Code or Micro QR Code, this functionality is also provided by Microsoft to developers through the plugin `Microsoft.MixedReality.QR` present in the NuGet package manager, however it comes with some problems.

  Two of them being surrounded by the fact that the plugin works around the embedded code functionality of the HL2 and all detected code are stored in the same "memory" and as such

shared between the device and the application, the other being that the "historic" of detected codes is only cleaned when the HL2 is restarted and its independent of the application.

Figure 6.20: Expected Behaviour of QR Code Reader at Passive Mode in Mixed Reality [Author's Source]

A more worrisome problem is that is not possible to reset, edit or filter the historic of detection's by code. Taking in account the mentioned problems it was searched others methods, since part of goal was to minimize overwork in the device, it was developed the MS `QR Code Decode` (6.4.2), the `QR Code Reader` is then responsible to manage the capturing of frames and sent them through a WS to decode while avoiding conflicts in using `Media Capture` and then act hover the code content.

The current implementation is limited to an "one-shot" detection, this is, it does not work in a passive mode, being already able to interpret data, however a present problem is the lack of a feature of the plugin from Microsoft, which is the ability to locate the QR Code in 3D as well to detect rotation and scale, no solution was found for the scale problem, however it was reused a method used during one of the trials to do `Face and Emotion Recognition` to determine the 3D position, however it only works in codes located in surfaces equal or larger than a laptop screen, since that prejudices this feature it was decided to list the detected codes in a panel when used the "one-shot" method, while the detection in the background

would focus in the codes that are focused in being recognised by the application and which should be located in walls, floors or as shown in Figure 6.20.

- **Notifications:** One of the main features expected in any kind of application that uses real-time data is notifications, while Microsoft provides a already a similar feature called `Dialog` through MRTK this is but a UI component to be toggled, the focus of this feature however is the creation of timed notifications, for example to alert a nurse to give medication (Figure 6.21), this features involves getting the information about necessary alerts in the DB specifically for the logged user and create timed events based on `coroutines`, a specific behaviour of this feature, following the mentioned example is to when done `Face and Emotion Recognition` also return a visual indicator that the detected patient is also the one expected to take the medication.

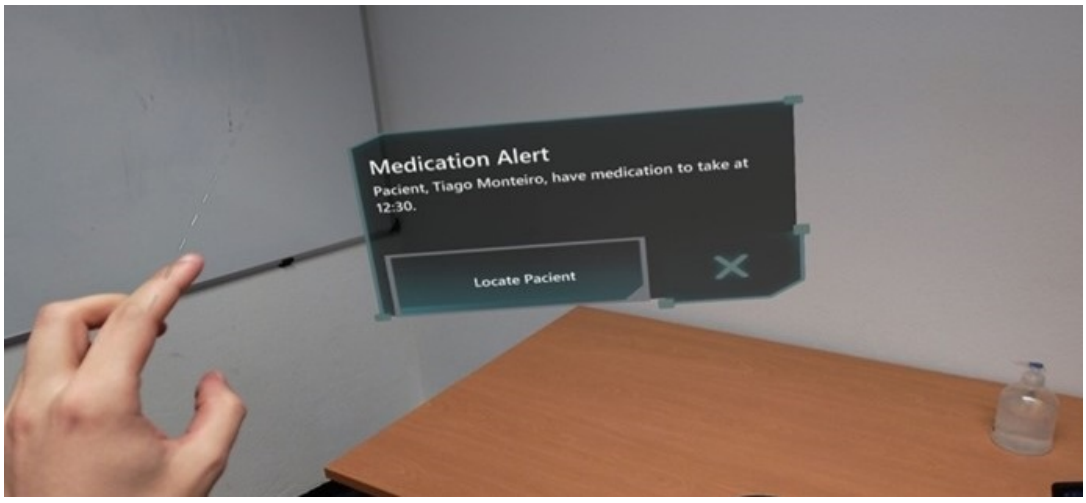

Figure 6.21: Medication Alert in Mixed Reality [133]

While not implemented and not planned to, an interesting behaviour would be a possibility to when notified about the need of giving medication also provide the possibility of starting a navigation system to guide the nurse till the patient, for situations where the patient is free to move, however this would require IoT devices.

- **Face and Emotion Recognition:** This feature is the principal goal defined for the MR application, being also the hardest component of the project and the more time consuming,

it consists in the ability of the user of a MR HMD, in this case the HL2, to recognise patients and their emotional state, displaying a visual marker over them (Figure 6.22), being the icons used to represent the emotions a modified version from the package by Anna Vidal [134].

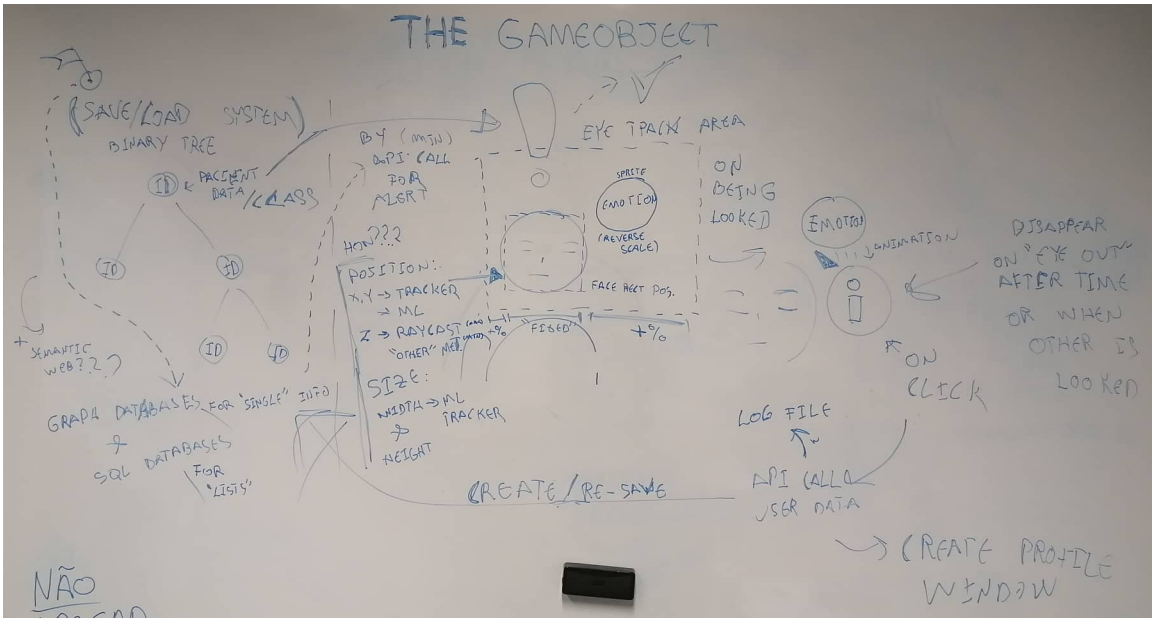

Figure 6.22: Face and Emotion Recognition Visual Marker (11/10/2022)

The current state of this component is the most closed to a final product being pendent of the finalisation of the MS `Face Detection with YuNet` (6.4.2), while passing by many changes, some of these moments can be highlighted as stages of progression.

- **First Stage:** In this stage the application would capture the frame through the class `Media Capture`, encode it to Base64 and sent it in JSON to the server for inference under a HTTP request, during this stage it was used the FastAPI API and the inference was done using DeepFace with OpenCV being afterwards sent the bounding box to the application, at this moment the emotion or face recognition was not the focus, in the application a similar approach of the application Lab Assist Vision [56] was done, however this applications relies in using Azure Services to do the inference, requiring an adaptation which was also complicated since most of the related code is under directives and is not interpreted by the IDE (6.1).

Another notable problem was the acquisition of `SpatialCoordinateSystem`, a necessary object which represents a coordinate system and is used to reason about the user's surroundings, the known method was already deprecated and no working alternative was found in the documentation, however it was found a working solution (Listing 6.5.2) among the many discussions in the developers forums.

```
1 private static SpatialCoordinateSystem CreateWorldOrigin() {
2     SpatialCoordinateSystem origin = PerceptionInterop.
    GetSceneCoordinateSystem(UnityEngine.Pose.identity) as
    SpatialCoordinateSystem;
3     return origin;
4 }
```

Listing 6.2: Getting SpatialCoordianeSystem (World Origin) in Unity

This methodology did not work since the application requires the user to stand still between the moment of capturing the frame till receiving the bounding box's data, being used OpenCV to create trackers of the objects (faces), this was implemented since the author was unknown of this particularity of CV at that time, being expected that CV would be able to recognise the objects in further frames by saving the one of which was done the inference.

– **Second Stage:** Since the use of OpenCV Trackers was not an acceptable approach it was decided to do the tracking by making consecutive requests of inferences to the server, this could not be done using HTTP Requests since would overwork the server as such it was adapted to use WS instead. While feasible the time taken between each inference was to big to create a real-time sensation and it was searched different approaches.

  In the meantime it was started the use of Face Recognition however the `embeddings` were saved in a file in the server and it was also by this moment that it was decided the refactoring of the Web API and the migration of certain components to a cloud-based server, during that process were created the MS and tested the use of direct requests to

a gRPC MS since it was not possible, instead of sending the frames in Base64 it was changed to Protobuf, this changes diminished the time but was still unfeasible.

– **Second to Third Stage**: After done the necessary adaptations, while not fulfilling the needs, it was detected that it also didn't marked the objects correctly, that is, the position was decided by the hitpoint of a Raycast against the Spatial Mesh, being the direction of the RayCast correspondent to the center of the bounding box, which was converted to a 3D position. While working under the presented scenario of the project, since the objects were located over a surface that was detected by the Spatial Mapping of the HL2, this is, in a screen, the same did not applied to persons and as such the visual mark was created far behind the detected person.

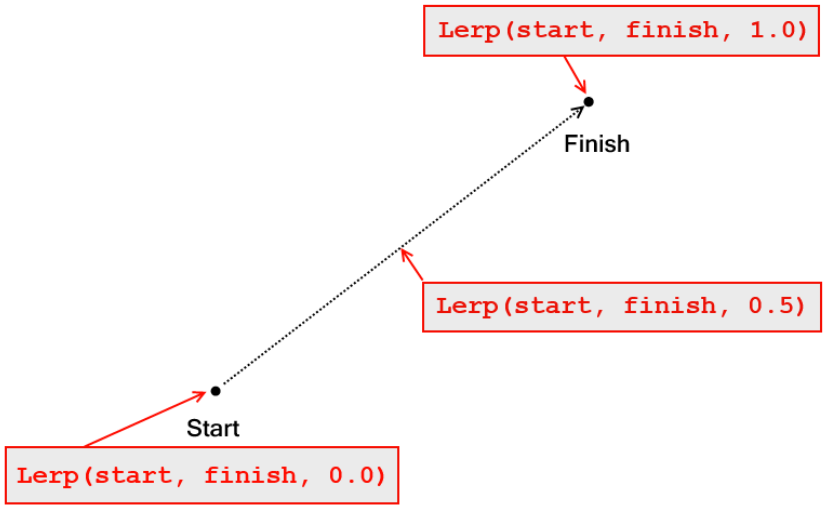

Figure 6.23: Lerp Concept [135]

As solution it was used part of the sample provided by Microsoft for Face Tracking using `FaceAnalysis API` from Azure and the WinRT class `FaceTracker`, this sample was also used for trials however for some undiscovered reason, while getting a distance close to the real world, the direction was random, another problem is that it was also limited to one face.

Each method executes by itself, getting a distance and direction separately, afterwards the results are merged in order to get a 3D position closer to reality, this method returns

the position on the axis between the User and the Hit Point with the Spatial Mesh and the same distance to the User as the distance between this and the position resulting from the second method, similarly with the method `Lerp` (Figure 6.23) provided by Unity. resulting as shown in Figure 6.24.

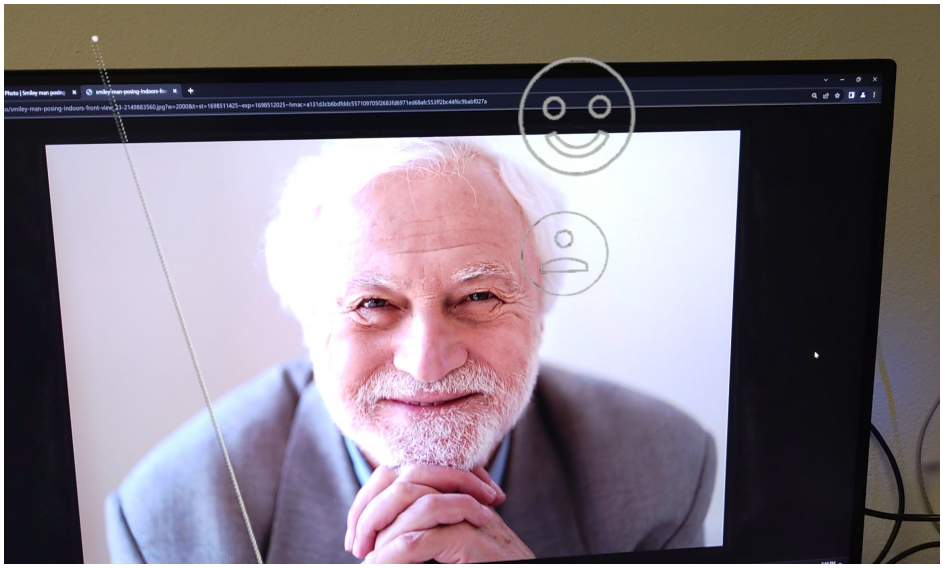

Figure 6.24: Face Detection of [136] in Second to Third Stage [Author's Source]

– **Third Stage:** This stages resumes in abandoning the Face Detection, and consecutively the "tracking", in the server and bring that responsibility to the application, it was tried the use of Face Detection with the Unity asset OpenCV For Unity, adapting a existent example to use the camera of the HL2. This example makes use of both LPB and HaaR, the last as being more accurate but heavy runs in a secondary thread through the use of Task's and not consecutively, being used mainly to detect faces while LPB, which is less accurate but faster, runs consecutively and takes the bounding box's from HaaR as area of focus.

  While doing the tracking and keeping also a historic of positions of the bounding box's, the frame and the bounding box's were also sent to the server for Face Recognition, creating a snapshot of the bounding box's at that time for validation when returned the results from the Face Recognition. This flow also worked however HL2 was unable to support more than ~2 minutes before shutting down, as an attempt to decrease the

overwork of the HL2 the part done by the HaaR was instead change to be done in the server but the problem persisted.

– **Fourth Stage:** So far the used methods were cross-platform, however since the HMD was unable to support them it was decided to use methods from the WinRT, in this library there's the class `FaceDetector` which uses a private model from Microsoft to detect faces, having high accuracy and speed while not overworking the device, the adaptation of this method proved hard since it was not interpreted by the IDE and to test it was necessary to build the application first.

  Fortunately during August 2022 it was published by Matthew Corbett in GitHub [137] a simple yet working project using `FaceDetector` in HL2 which was used as base for refactoring this feature to the current state of the project and while the methods developed during the `Second to Third Stage` were no more needed for this feature they found new usability for the `QR Code Reader` (6.5.2).

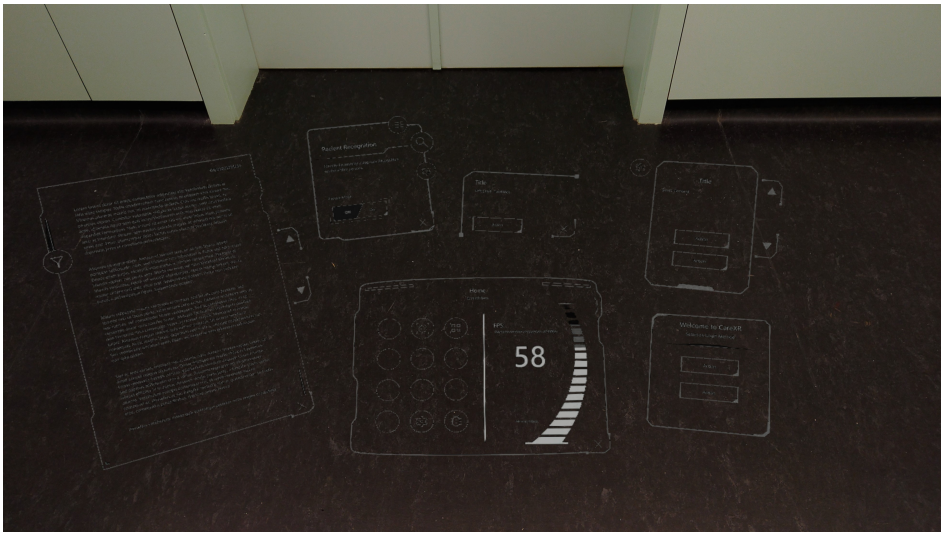

Figure 6.25: Samples of Mixed Reality Interfaces [Author's Source]

- **Interface Handler:** This is a very particular feature developed with the HL2 in mind and the Spatial Computing ideology in account, firstly, when developing interfaces for Unity applications the interface is commonly developed by using `sprites`, which are easily created by converting normal raw images, however HL2, when used MRTK2, is unable to render

sprites, provoking situations where only one of the eyes can see it or none at all. Instead it uses materials and the interface is treated as a normal GameObject instead of Canvas, and a similar problem happens with blocks of text if not hover a material.

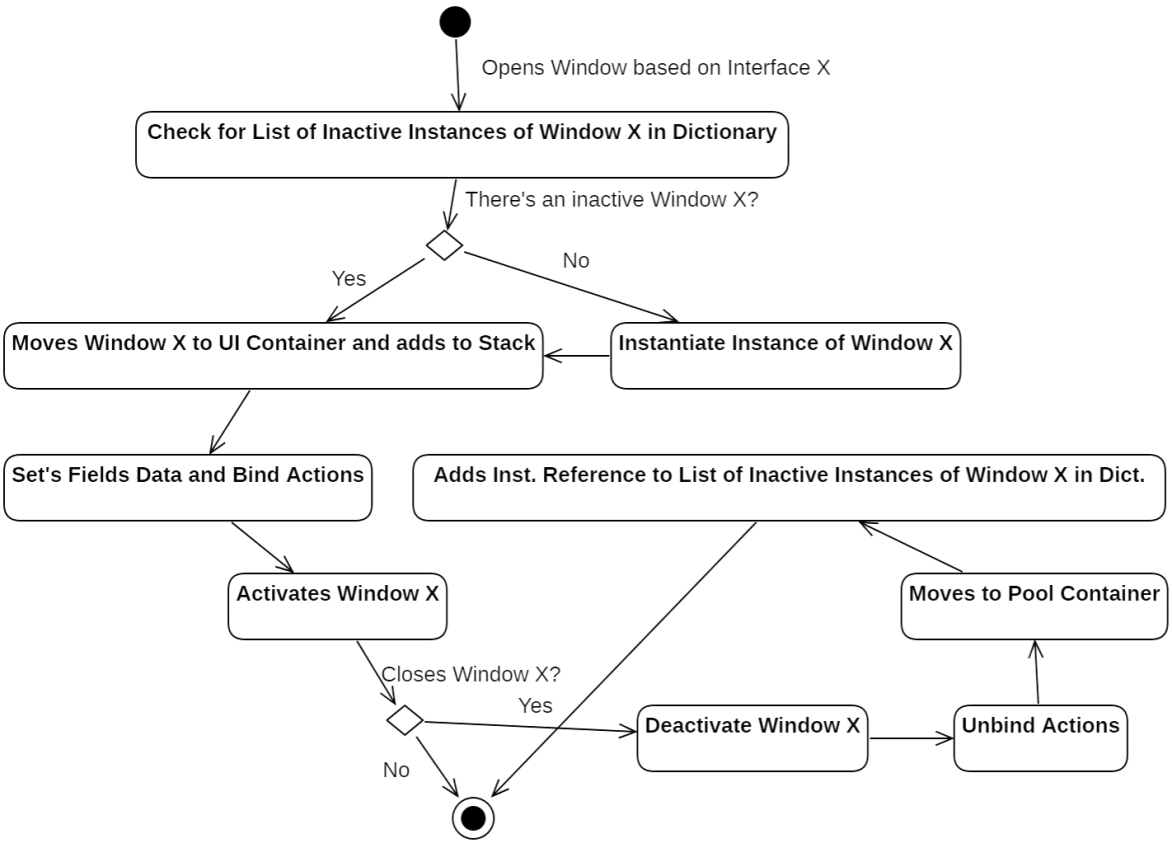

Figure 6.26: Activity Diagram of Interface Pooling [Author's Source]

However this comes with some limitations, being hard to manipulate the interfaces inside of the engine, instead if necessary the same type of interface with different scales, and not using the same aspect ratio, they must be created a priori, which creates the problem in managing the existent interfaces in-game, being then developed a set of ScriptableObjects which store arrays of structs being converted in dictionaries when started the application, a pair of ScriptableObjects store the references to the materials of different kind of buttons and states, simulating some common features of the component Button in Unity, one for Round Buttons and the other for Regular Buttons.

A third `ScriptableObject` was prepared (Appendix D.3), this stores references to each existent interface, as well manipulable content, such as texts blocks and buttons, these different contents are distinguished by using `enums`, while the interface identifier is also distinguished by an `enum`.

The second aspect is related to the concept of Spatial Computing, when developing games or serious games, the necessary interfaces are commonly already predefined as well the expected navigation, for example, the user is unable to have is own profile opened twice, this is the normal and expected behavior enabled by Unity.

However with Spatial Computing the interfaces resemble more websites, where the user may be able to have *N* pages of their profile opened, a probable approach to solve the situation would be the instantiation of the interface and when no more needed the deactivation or destruction, however both method are bad for performance, a `GameObject`, even if deactivated still consume resources of the device, while the methods of Instantiate and Destroy `GameObject` are among the heaviest methods of Unity.

Instead it was implemented the principle of Pooling (Figure 6.26) but applied to interfaces, which is basically the reuse of `GameObjects`, which means that when a certain interface is closed, she is putted on a pool, and when a interface with the same identifier, based on the `dictionary` from the `ScriptableObject`, is needed she is reactivated and the content is updated.

A different concept applied is the layers, on contrary to predefined flows with references, the UI Instances are saved in `Stacks` enabling the user to navigate back without complications. The current interface is a modified version of [138].

6.5.3 Web Application

The developed application focus in functionalities implementation since the logic is independent of how a UI is done, look alike, or the UX, being only dependent of the necessary information to be provided by the user, the current interface is based on the template resulted from a tutorial in React.js and Material UI [139].

React.js is well known in the developers community for being sensible in relation to the packages versions as such in the Table 6.1 can be seen the principal packages in the project and their current versions.

Table 6.1: Principal React.js Packages in Use

Package Name	**Version**	**Use**
apollo/client	3.7.14	Tools to work with GraphQL
fortawesome/fontawesome	1.1.8	Icons and CSS
mui/material	5.12.3	UI Components
axios	1.4.0	HTTP Requests
formik	2.2.9	Build Forms
graphql	16.6.0	GraphQL Query Formatter
qrcode	1.5.3	QR Code Generator
react	18.2.0	-
react-auth-kit	2.12.2	Authentication Management
react-pro-sidebar	0.7.1	Sidebar Facilitator
react-redux	8.0.5	Store states
react-router-dom	6.11.1	-
react-scripts	5.0.1	-
react-use-websocket	4.3.1	Use of Websockets
redux-persist	6.0.0	Save Redux states in AsyncStorage
uuid	9.0.0	UUID Generator

- **QR Code Authentication Displayer:** This functionality (Appendix E.1.3) when activated by the user establishes a WS connection by using the package `react-use-websocket`, while requesting from the Web API a JWT through a normal GET request which contains a channel UUID, the same UUID also is sent outside of the JWT and used to define the room in the WS, the token itself is displayed in the screen with the use of the package `qrcode` to be read by the MR application (6.5.2).

- **VR Session Control Panel:** This feature refers to a section of the Web Application (Appendix E.1.2) that works as a dashboard to control VR sessions in the VR application, being the goal minimize the necessity of changing the HMD between users to not become uncomfortable the situation as to enable sessions at long distance.

When opened this section is required a NanoID to start the process described in 6.4.3 and after established a connection between the Web Application and the VR Application its presented to the user the option to start a 3D Model based exercise or a Panoramic Image (360º) based exercise.

Being defined as goal exercises of eye tracking in panoramic images, the user can after selecting "Panoramic Images Exercises" select one from a list that consists in a Data Grid component from MUI with data retrieved from a Web API request, these are limited to only Panoramic Images associated with the Institution to which the user belongs.

```
<Box position="relative">
{showLog == false ? <>
<img src={`${props.panoramicData["base64"]}`} alt="Panoramic" width
   ={"92%"}/>
    {props.panoramicData["mapping"].map((box) => (
<Box key={box.uuid}
    sx={{ position: 'absolute',
    left: `${((box.size.x / imgRatio) / 37.4).toFixed(2)}%`,
    top: `${((box.size.y / imgRatio) / 17.3).toFixed(2)}%`,
    width: `${((box.size.width / imgRatio) / 40).toFixed(2)}%`,
    height: `${((box.size.height / imgRatio) / 17.5).toFixed(2)}%`,
    border: `2px solid ${selected === box.uuid ? '#1bfc06' : '#FF0000
   '}`,
}} /> ))} </> :
```

Listing 6.3: Display of a Panoramic Image and drawing of Bounding Box's in React.js

When selected a Panoramic Image, it will be transferred to the Web Application being displayed as a simple 2D image, while defined bounding box's are drawn hover the image (Listing 6.5.3), the same image is also transferred to VR Application being processed as described in 6.5.1.

After done the loading of the image the user can select the type of exercise and define a label/question, for example "Where is your son?", and when started a exercise based in

`Recognition Ability with Eye Tracking Exercise` the user can see which object is being looked at through the data received from the SSE `VR Panoramic Session Data Receiver` (Figure 6.27) while being able to give some basic commands.

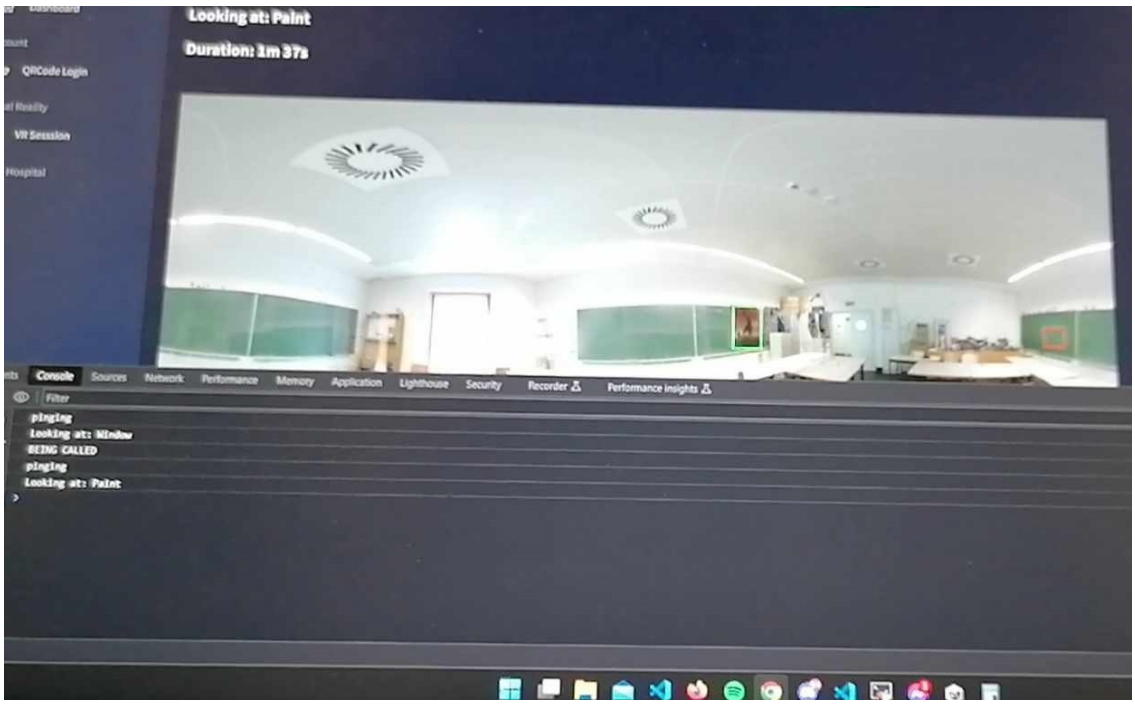

Figure 6.27: Bounding Box Highlight in Exercise with Panoramic Image (Test Image) [Author's Source]

This is done by developing a main component that stores WS related data like the variable containing the connection as well the state of the session, this data would then be passed for others components as `props`, each of these components contains the interface of each state of the session and the specific logic, these child components will then be rendered conditionally based on the session state (disconnected, connected, loading, running).

Chapter 7 Conclusions

At the current moment of the thesis submission, the project GreenHealth its still on going, being the submission needed for academic progression, and the development itself continues, the VR application is considered completed since it already surpass the necessaries requirements, however the MR application is still lacking in some features due to the current state of development of the MS `Face Detection with YuNet`.

The work done contributed hugely for self-improvement as Full-Stack developer, thanks to the possibility to be introduced on a whole new range of technologies which were new for the author/developer, some even unheard till this moment. It was also a big step as researcher due to the need to know about unfamiliar fields such as ML, CV and the Health Industry.

The current project resulted in a paper titled "Using Mixed Reality and Machine Learning to Assist Caregivers in Nursing Home and Promote Well-being" [140]. The article was submitted to the "CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2022" and has been accepted and published with open access in the journal Procedia Computer Science as part of the conference proceedings (Appendix B.1) being presented a more advanced conceptual model.

Additionally, a book chapter titled "Intangible Approaches to Improve Individual Health Indicators and Empower Caregivers" [133] was also submitted and accepted. Being published in the collected work titled "Internet of Everything for Smart City and Smart Healthcare Applications," edited by Nishu Gupta and Sumita Mishra, and under the imprint Springer being part of the book series "Signals and Communication Technology" (Appedinx B.1). The chapter includes a summary of the components developed till the moment of submission of the chapter, which

are discussed in more details in this thesis. It's worth noting that improvements have been made during the course of the project since then.

7.1 Future Work

While this project had 18 months for realization, taking in attention the learning processes, the time to acquisition of certain equipment and degree classes which involved traveling between places, many components of the project are still lacking and can be considered for future improvements.

- **Primary Database:** The Primary DB, being it the Graph DB, was designed mostly to support the applications and to analyse potential uses of DS and Data Analysis, however it should be refactored to follow the FHIR standard, this would require a deep analyse and study of the standard and a partnership with different medical institutions would also be a good approach to test the interoperability. Since the standard lacks in terms of tools and technologies recommendations the use of Neo4j would still be a good choice;

- **Web Application:** Another component developed mostly for support, the focus was the implementation of the features, a review in terms of UI and UX is necessary, also further development to implement "common" features like CRUD operations in the DB and to be able to be used by different end-users, this would also require a better understatement of the mentioned standard and the definition of requirements with institutions;

- **VR Application:** The current VR application is only prepared to execute exercises involving "object recognition" in Panoramic Images, however others exercises could be developed like analysing the interest of the patient or even learning processes as well the implementation of the stereoscopic concept, and although more difficult, exercises involving 360º videos. Exercises involving 3D models would also prove a good addition, if able to reproduce exercises like the mentioned ones in the State of the Art (3), however it would involve the development of Multiplayer Servers, like with Photon or FishNet;

- **MR Application:** This component was the one which took more time and like the Web Application, the development was focused on specific features, requiring further developments

to comply with the requirements of institutions. A migration from the current used MRTK2 to MRTK3 should also be done;

- **AR Application:** No AR application was developed;
- **Computer-Based Applications:** Some aspects of the project can't be answered by a Web-Based application or Unity, the main one detected being the upload of Panoramic Images, due to the limit size of uploads which would affect the image quality and Unity being unable to do such thing, being necessary the development of a computer application like with UWP or Windows Forms. If developed exercises in the VR application involving 3D models, it may be developed personalised exercises, which would be created by the development team and then integrated in the service, or it can be provided the ability to the institutions to create exercises by using pre-defined assets and 3D objects, this would require a desktop Unity application and would involve mechanics like the ones present in the game "Meet Your Maker";
- **Back-End:** The current back-end works with the implemented features, however improvements in the others components would require further development, in terms of the VR application and the exercises, it would probably require a refactoring of the WS services to better maintenance and scalability;

Aside from the components development, its also required the test of the developed prototypes, as referred previously, some necessary components related with the MR use cases are still in development, however, while the VR is in a state ready to test, due to the recent pandemic that had a big effect in elders and nursing homes it was not possible to find collaborators, the fact of involving sensible and private data also proved being a constraint. It is hoped to have the opportunity to test the current prototypes in a close future.

References

Bibliography

[1] M. Ulfa, M. Azuma, and A. Steiner, "Burnout status of healthcare workers in the world during the peak period of the COVID-19 pandemic", en, *Front Psychol*, vol. 13, p. 952 783, Sep. 2022. DOI: 10.3389/fpsyg.2022.952783.

[2] J. A. Delmerico, R. Poranne, F. Bogo, H. Oleynikova, E. Vollenweider, S. Coros, J. I. Nieto, and M. Pollefeys, "Spatial computing and intuitive interaction: Bringing mixed reality and robotics together", *CoRR*, vol. abs/2202.01493, 2022. DOI: arXiv:2202.01493. arXiv: 2202.01493.

[3] J. Dalton, *Reality check: How immersive technologies can transform your business*. Kogan Page Publishers, Jan. 2021.

[4] A. Çöltekin, I. Lochhead, M. Madden, S. Christophe, A. Devaux, C. Pettit, O. Lock, S. Shukla, L. Herman, Z. Stachoň, P. Kubíček, D. Snopková, S. Bernardes, and N. Hedley, "Extended reality in spatial sciences: A review of research challenges and future directions", *ISPRS International Journal of Geo-Information*, vol. 9, no. 7, 2020, ISSN: 2220-9964. DOI: 10.3390/ijgi9070439.

[5] P. A. Rauschnabel, R. Felix, C. Hinsch, H. Shahab, and F. Alt, "What is xr? towards a framework for augmented and virtual reality", *Computers in Human Behavior*, vol. 133, p. 107 289, 2022, ISSN: 0747-5632. DOI: 10.1016/j.chb.2022.107289.

[7] M. H. Shahab, E. Ghazali, and M. Mohtar, "The role of elaboration likelihood model in consumer behaviour research and its extension to new technologies: A review and future research agenda", *International Journal of Consumer Studies*, vol. 45, no. 4, pp. 664–689, 2021. DOI: 10.1111/ijcs.12658.

[13] K. E. Schein and P. A. Rauschnabel, "Augmented reality in manufacturing: Exploring workers' perceptions of barriers", *IEEE Transactions on Engineering Management*, pp. 1–14, 2021. DOI: 10.1109/TEM.2021.3093833.

[14] Y.-C. Tan, S. R. Chandukala, and S. K. Reddy, "Augmented reality in retail and its impact on sales", *Journal of Marketing*, vol. 86, no. 1, pp. 48–66, 2022.

[15] M. Billinghurst, A. Clark, and G. Lee, "A survey of augmented reality", *Foundations and Trends® in Human–Computer Interaction*, vol. 8, no. 2-3, pp. 73–272, 2015, ISSN: 1551-3955. DOI: 10.1561/1100000049.

[16] J. Liu, Y. Xie, S. Gu, and X. Chen, "A slam-based mobile augmented reality tracking registration algorithm", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 01, p. 2 054 005, 2020. DOI: 10.1142/S0218001420540051.

[17] P. A. Rauschnabel, "Virtually enhancing the real world with holograms: An exploration of expected gratifications of using augmented reality smart glasses", *Psychology & Marketing*, vol. 35, no. 8, pp. 557–572, 2018. DOI: 10.1002/mar.21106.

[19] W. D. Hoyer, M. Kroschke, B. Schmitt, K. Kraume, and V. Shankar, "Transforming the customer experience through new technologies", *Journal of Interactive Marketing*, vol. 51, pp. 57–71, 2020, Special issue on Big Data, Technology-Driven CRM & Artificial Intelligence, ISSN: 1094-9968. DOI: 10.1016/j.intmar.2020.04.001.

[20] M. Doughty, N. R. Ghugre, and G. A. Wright, "Augmenting performance: A systematic review of optical see-through head-mounted displays in surgery", *Journal of Imaging*, vol. 8, no. 7, 2022, ISSN: 2313-433X. DOI: 10.3390/jimaging8070203.

[25] H. G. Hoffman, W. J. Meyer, M. Ramirez, L. Roberts, E. J. Seibel, B. Atzori, S. R. Sharar, and D. R. Patterson, "Feasibility of articulated arm mounted oculus rift virtual reality goggles for adjunctive pain control during occupational therapy in pediatric burn patients", *Cyberpsychology, Behavior, and Social Networking*, vol. 17, no. 6, pp. 397–401, 2014, PMID: 24892204. DOI: 10.1089/cyber.2014.0058.

[27] M. Xu, J. David, and S. Kim, "The fourth industrial revolution: Opportunities and challenges", *International Journal of Financial Research*, vol. 9, p. 90, Feb. 2018. DOI: 10.5430/ijfr.v9n2p90.

[28] D. Agbaji, B. Lund, and N. R. Mannuru, *Perceptions of the fourth industrial revolution and artificial intelligence impact on society*, 2023. arXiv: 2308.02030 [cs.CY].

[29] R. Rayan and C. Tsagkaris, "Iot technologies for smart healthcare", in. Dec. 2021, pp. 29–46, ISBN: 9781003207856. DOI: 10.1201/9781003207856-3.

[30] D. V. Dimitrov, "Medical internet of things and big data in healthcare", en, *Healthc Inform Res*, vol. 22, no. 3, pp. 156–163, Jul. 2016. DOI: 10.4258/hir.2016.22.3.156.

[31] P. Rashidi and A. Mihailidis, "A survey on ambient-assisted living tools for older adults", *IEEE journal of biomedical and health informatics*, vol. 17, pp. 579–90, May 2013. DOI: 10.1109/JBHI.2012.2234129.

[32] M. Al-rawashdeh, P. Keikhosrokiani, B. Belaton, M. Alawida, and A. Zwiri, “Iot adoption and application for smart healthcare: A systematic review”, *Sensors*, vol. 22, p. 5377, Jul. 2022. DOI: 10.3390/s22145377.

[33] B. Farahani, F. Firouzi, and K. Chakrabarty, “Healthcare iot”, in *Intelligent Internet of Things: From Device to Fog and Cloud*, F. Firouzi, K. Chakrabarty, and S. Nassif, Eds. Cham: Springer International Publishing, 2020, pp. 515–545, ISBN: 978-3-030-30367-9. DOI: 10.1007/978-3-030-30367-9_11.

[36] J. Lenoir, “Effective user interface of iot system at nursing homes”, in *HCI International 2019 - Posters*, C. Stephanidis, Ed., Cham: Springer International Publishing, 2019, pp. 490–498, ISBN: 978-3-030-23528-4. DOI: 10.1007/978-3-030-23528-4_67.

[40] J. K. Ramjist, N. Coburn, D. R. Urbach, A. Govindarajan, K. A. Armstrong, A. L. Scott, and N. N. Baxter, “Disposal of Paper Records Containing Personal Information in Hospitals”, *JAMA*, vol. 319, no. 11, pp. 1162–1163, Mar. 2018, ISSN: 0098-7484. DOI: 10.1001/jama.2017.21533.

[42] P. Wicks, J. Stamford, M. Grootenhuis, L. Haverman, and S. Ahmed, “Innovations in e-health”, *Quality of life research : an international journal of quality of life aspects of treatment, care and rehabilitation*, vol. 23, Jul. 2013. DOI: 10.1007/s11136-013-0458-x.

[45] T. Heart, O. Ben-Assuli, and I. Shabtai, “A review of phr, emr and ehr integration: A more personalized healthcare and public health policy”, *Health Policy and Technology*, vol. 6, no. 1, pp. 20–25, 2017, ISSN: 2211-8837. DOI: 10.1016/j.hlpt.2016.08.002.

[46] S. Rakesh Kumar, N. Gayathri, S. Muthuramalingam, B. Balamurugan, C. Ramesh, and M. Nallakaruppan, “Chapter 13 - medical big data mining and processing in e-healthcare”, in *Internet of Things in Biomedical Engineering*, V. E. Balas, L. H. Son, S. Jha, M. Khari, and R. Kumar, Eds., Academic Press, 2019, pp. 323–339, ISBN: 978-0-12-817356-5. DOI: 10.1016/B978-0-12-817356-5.00016-4.

[47] R. Koppel and C. U. Lehmann, “Implications of an emerging EHR monoculture for hospitals and healthcare systems”, *Journal of the American Medical Informatics Association*, vol. 22, no. 2, pp. 465–471, Oct. 2014, ISSN: 1067-5027. DOI: 10.1136/amiajnl-2014-003023.

[48] D. F. Sittig, Ed., *Electronic Health Records: Challenges in Design and Implementation*, en, 0th ed. Apple Academic Press, Nov. 2013, ISBN: 9780429158735. DOI: 10.1201/b16306.

[51] C. N. Vorisek, M. Lehne, S. A. I. Klopfenstein, P. J. Mayer, A. Bartschke, T. Haese, and S. Thun, “Fast healthcare interoperability resources (fhir) for interoperability in health research: Systematic review”, *JMIR Med Inform*, vol. 10, no. 7, e35724, Jul. 2022, ISSN: 2291-9694. DOI: 10.2196/35724.

[52] M. A. Kramer and C. Moesel, “Interoperability with multiple Fast Healthcare Interoperability Resources (FHIR®) profiles and versions”, *JAMIA Open*, vol. 6, no. 1, ooad001, Feb. 2023, ISSN: 2574-2531. DOI: 10.1093/jamiaopen/ooad001.

[53] J. Nan and L.-Q. Xu, "Designing interoperable health care services based on fast healthcare interoperability resources: Literature review", *JMIR Med Inform*, vol. 11, e44842, Aug. 2023, ISSN: 2291-9694. DOI: 10.2196/44842.

[55] S. Singh and F. Nasoz, "Facial expression recognition with convolutional neural networks", in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020, pp. 0324–0328. DOI: 10.1109/CCWC47524.2020.9031283.

[56] S. Pereto, "Mixed reality task assistance for pharmaceutical laboratories using markerless object tracking and cloud services", masterThesis.

[57] M. Blusi and J. C. Nieves, "Feasibility and acceptability of smart augmented reality assisting patients with medication pillbox self-management", en, *Stud Health Technol Inform*, vol. 264, pp. 521–525, Aug. 2019.

[58] S. Sirilak and P. Muneesawang, "A new procedure for advancing telemedicine using the hololens", *IEEE Access*, vol. 6, pp. 60 224–60 233, 2018. DOI: 10.1109/ACCESS.2018.2875558.

[59] L. Qian, C. Song, Y. Jiang, Q. Luo, X. Ma, P. W. Chiu, Z. Li, and P. Kazanzides, "Flexivision: Teleporting the surgeon's eyes via robotic flexible endoscope and head-mounted display", in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3281–3287. DOI: 10.1109/IROS45743.2020.9340716.

[60] F. Heinrich, F. Huettl, G. Schmidt, M. Paschold, W. Kneist, T. Huber, and C. Hansen, "HoloPointer: A virtual augmented reality pointer for laparoscopic surgery training", *International Journal of Computer Assisted Radiology and Surgery*, vol. 16, no. 1, pp. 161–168, Jan. 2021. DOI: 10.1007/s11548-020-02272-2.

[61] T. Jung, N. Moorhouse, X. Shi, and M. F. Amin, "A virtual Reality-Supported intervention for pulmonary rehabilitation of patients with chronic obstructive pulmonary disease: Mixed methods study", en, *J Med Internet Res*, vol. 22, no. 7, e14178, Jul. 2020. DOI: 10.2196/14178.

[62] S. Rocha and A. Lopes, "Navigation based application with augmented reality and accessibility", in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '20, Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–9, ISBN: 9781450368193. DOI: 10.1145/3334480.3383004.

[63] R. E. Brimelow, B. Dawe, and N. Dissanayaka, "Preliminary research: Virtual reality in residential aged care to reduce apathy and improve mood", *Cyberpsychology, Behavior, and Social Networking*, vol. 23, no. 3, pp. 165–170, 2020. DOI: 10.1089/cyber.2019.0286. eprint: 10.1089/cyber.2019.0286.

[64] G. Goodall, I. Ciobanu, K. Taraldsen, J. Sørgaard, A. Marin, R. Draghici, M.-V. Zamfir, M. Berteanu, W. Maetzler, and J. A. Serrano, "The use of virtual and immersive technology in creating personalized multisensory spaces for people living with dementia (sense-garden): Protocol for a multisite before-after trial", *JMIR Res Protoc*, vol. 8, no. 9, e14096, Aug. 2019, ISSN: 1929-0748. DOI: 10.2196/14096.

[65] W. Moyle, C. Jones, T. Dwan, and T. Petrovich, "Effectiveness of a virtual reality forest on people with dementia: A mixed methods pilot study", *The Gerontologist*, vol. 58, no. 3, pp. 478–487, Mar. 2017, ISSN: 0016-9013. DOI: 10.1093/geront/gnw270.

[66] I. Ciobanu, R. Draghici, A. Marin, M.-V. Zamfir, A. Iliescu, M. Zamfir, G. Goodall, M. Popescu, J. A. Serrano, and M. Berteanu, "Case series of an ICT-based multimodal intervention program for people with major neurocognitive disorders: The SENSE-GARDEN project", *Gerontology and Geriatric Medicine*, vol. 8, no. 3, pp. 1–6, Jul. 2022. DOI: 10.24966/ggm-8662/100137.

[67] M. Ilioudi, P. Lindner, L. Ali, S. Wallström, A. O. Thunström, M. Ioannou, N. Anving, V. Johansson, W. Hamilton, Ö. Falk, and S. Steingrimsson, "Physical versus virtual reality–based calm rooms for psychiatric inpatients: Quasi-randomized trial", *J Med Internet Res*, vol. 25, e42365, May 2023, ISSN: 1438-8871. DOI: 10.2196/42365.

[68] S. Masoumzadeh and Z. Moussavi, "Does practicing with a virtual reality driving simulator improve spatial cognition in older adults? a pilot study", *Neuroscience Insights*, vol. 15, p. 2 633 105 520 967 930, 2020, PMID: 35174332. DOI: 10.1177/2633105520967930.

[69] L. D. Deusdado and A. F. Antunes, "Virtual reality haptic device for mental illness treatment", *Procedia Computer Science*, vol. 219, pp. 1112–1119, 2023, CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2022, ISSN: 1877-0509. DOI: 10.1016/j.procs.2023.01.391.

[70] R. P. Lopes, B. Barroso, L. Deusdado, A. Novo, M. Guimarães, J. P. Teixeira, and P. Leitão, "Digital technologies for innovative mental health rehabilitation", *Electronics*, vol. 10, no. 18, 2021, ISSN: 2079-9292. DOI: 10.3390/electronics10182260.

[71] D. Martins, M. Neves, B. Marques, S. BráS, and J. M. Fernandes, "Immerse yourself in a fight against your fears: A vision for using virtual reality serious games and physiology assessment in phobia treatment", in *Proceedings of the 22nd International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '23, New York, NY, USA: Association for Computing Machinery, 2023, pp. 477–479, ISBN: 9798400709210. DOI: 10.1145/3626705.3631781.

[72] E. Malbos, G. H. Burgess, and C. Lançon, "Virtual reality and fear of shark attack: A case study for the treatment of squalophobia", *Clinical Case Studies*, vol. 19, no. 5, pp. 339–354, 2020. DOI: 10.1177/1534650120940014.

[73] X. Chen, L. Gong, A. Berce, B. Johansson, and M. Despeisse, "Implications of virtual reality on environmental sustainability in manufacturing industry: A case study", *Procedia CIRP*, vol. 104, pp. 464–469, 2021, 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0, ISSN: 2212-8271. DOI: 10.1016/j.procir.2021.11.078.

[74] S. K. Mishra, "Hospital overcrowding", en, *West J Med*, vol. 174, no. 3, p. 170, Mar. 2001. DOI: 10.1136/ewjm.174.3.170.

[75] D. Duong and L. Vogel, "Overworked health workers are "past the point of exhaustion"", en, *CMAJ*, vol. 195, no. 8, E309–E310, Feb. 2023. DOI: 10.1503/cmaj.1096042.

[76] N. Kshetri and Y. K. Dwivedi, "Pollution-reducing and pollution-generating effects of the metaverse", *International Journal of Information Management*, vol. 69, p. 102 620, 2023, ISSN: 0268-4012. DOI: 10.1016/j.ijinfomgt.2023.102620.

[77] B. Marques, A. Teixeira, S. Silva, J. Alves, P. Dias, and B. S. Santos, "A critical analysis on remote collaboration mediated by augmented reality: Making a case for improved characterization and evaluation of the collaborative process", *Computers & Graphics*, vol. 102, pp. 619–633, 2022, ISSN: 0097-8493. DOI: 10.1016/j.cag.2021.08.006.

[78] J. Cartucho, D. Shapira, H. Ashrafian, and S. Giannarou, "Multimodal mixed reality visualisation for intraoperative surgical guidance", *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, no. 5, pp. 819–826, May 2020. DOI: 10.1007/s11548-020-02165-4.

[82] J.-K. Chen and W.-Z. Lee, "An introduction of nosql databases based on their categories and application industries", *Algorithms*, vol. 12, no. 5, 2019, ISSN: 1999-4893. DOI: 10.3390/a12050106.

[83] S. Begum, "A Deeper Examination of NoSQL Database Models and Characteristics", *International Journal of Research Publication and Reviews*, pp. 1510–1514, Oct. 2022, ISSN: 25827421. DOI: 10.55248/gengpi.2022.3.10.56.

[84] W. Ali, M. U. Shafique, M. A. Majeed, and A. Raza, "Comparison between sql and nosql databases and their relationship with big data analytics", *Asian Journal of Research in Computer Science*, vol. 4, no. 2, pp. 1–10, Oct. 2019. DOI: 10.9734/ajrcos/2019/v4i230108.

[85] F. Michel, "Integrating heterogeneous data sources in the Web of data", Theses, Université Côte d'Azur, Mar. 2017. [Online]. Available: https://theses.hal.science/tel-01508602.

[87] R. Hecht and S. Jablonski, "Nosql evaluation: A use case oriented survey", in *2011 International Conference on Cloud and Service Computing*, 2011, pp. 336–341. DOI: 10.1109/CSC.2011.6138544.

[88] G. Kundu, N. Mukherjee, and S. Mondal, "Building a graph database for storing heterogeneous healthcare data", in *Information and Communication Technology for Intelligent Systems*, T. Senjyu, P. N. Mahalle, T. Perumal, and A. Joshi, Eds., Singapore: Springer Singapore, 2021, pp. 193–201. DOI: 10.1007/978-981-15-7062-9_19.

[90] M. Garvey and M. Jackson, "Introduction to object-oriented databases", *Information and Software Technology*, vol. 31, no. 10, pp. 521–528, 1989, ISSN: 0950-5849. DOI: 10.1016/0950-5849(89)90173-0.

[92] D. Ramesh, A. Sinha, and S. Singh, "Data modelling for discrete time series data using cassandra and mongodb", in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 2016, pp. 598–601. DOI: 10.1109/RAIT.2016.7507966.

[93] R. Liu and J. Yuan, "Benchmark time series database with iotdb-benchmark for iot scenarios", *CoRR*, vol. abs/1901.08304, 2019. DOI: 10.48550/arXiv.1901.08304. arXiv: 1901.08304.

[94] G. Brito and M. T. Valente, "Rest vs graphql: A controlled experiment", IEEE, Mar. 2020, pp. 81–91, ISBN: 978-1-7281-4659-1. DOI: 10.1109/ICSA47634.2020.00016.

[95] O. Hartig and J. Pérez, "Semantics and complexity of graphql", ACM Press, 2018, pp. 1155–1164, ISBN: 9781450356398. DOI: 10.1145/3178876.3186014.

[97] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment", *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019, ISSN: 2405-9595. DOI: 10.1016/j.icte.2017.12.005.

[98] P. Levchenko, D. Bankov, E. Khorov, and A. Lyakhov, "Performance comparison of nb-fi, sigfox, and lorawan", *Sensors*, vol. 22, no. 24, 2022, ISSN: 1424-8220. DOI: 10.3390/s22249633.

[100] S. Popić, D. Pezer, B. Mrazovac, and N. Teslić, "Performance evaluation of using protocol buffers in the internet of things communication", in *2016 International Conference on Smart Systems and Technologies (SST)*, 2016, pp. 261–265. DOI: 10.1109/SST.2016.7765670.

[106] S. I. Serengil and A. Ozpinar, "Lightface: A hybrid deep face recognition framework", in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Oct. 2020, pp. 1–5. DOI: 10.1109/ASYU50717.2020.9259802.

[107] ——, "Hyperextended lightface: A facial attribute analysis framework", in *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, Oct. 2021, pp. 1–4. DOI: 10.1109/ICEET53442.2021.9659697.

[112] S. Zaccardi, T. Frantz, D. Beckwée, E. Swinnen, and B. Jansen, "On-device execution of deep learning models on hololens2 for real-time augmented reality medical applications", *Sensors*, vol. 23, no. 21, Oct. 2023, ISSN: 1424-8220. DOI: 10.3390/s23218698.

[113] L. Lazar, "Neural Networks on Microsoft HoloLens 2", en, Bachelor Thesis, 2021. DOI: 10.18419/opus-11558.

[114] K. S.-Y. Yoon Byoung-Ha Kim Seon-Kyu, "Use of graph database for the integration of heterogeneous biological data", *Genomics Inform*, vol. 15, no. 1, pp. 19–27, 2017. DOI: 10.5808/GI.2017.15.1.19.

[115] X. Fang, L. Li, and Z. Wei, "Design of recommendation algorithm based on knowledge graph", *Journal of Physics: Conference Series*, vol. 2425, no. 1, p. 012 025, Feb. 2023. DOI: 10.1088/1742-6596/2425/1/012025.

[116] N. Petrovic, "Booking cancellation prediction relying on graph database in neo4j", IEEESTEC – 14th Student Projects Conference, Nov. 2021.

[117] A. Tammaa, "Mongodb case study on forbes", May 2022. DOI: 10.13140/RG.2.2.32766.46408.

[118] S. Di Martino, L. Fiadone, A. Peron, A. Riccabone, and V. N. Vitale, "Industrial internet of things: Persistence for time series with nosql databases", in *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2019, pp. 340–345. DOI: 10.1109/WETICE.2019.00076.

[122] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, *Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*, 2022. arXiv: 2207.02696 [cs.CV].

[123] W. Wu, H. Peng, and S. Yu, "YuNet: A tiny millisecond-level face detector", *Machine Intelligence Research*, vol. 20, no. 5, pp. 656–665, Oct. 2023. DOI: 10.1007/s11633-023-1423-y.

[126] R. Kosti, J. M. Alvarez, A. Recasens, and A. Lapedriza, "Context based emotion recognition using emotic dataset", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 11, pp. 2755–2766, Nov. 2020, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2019.2916866.

[127] M. Tomasetti, "An analysis of the performance of websockets in various programming languages and libraries", *SSRN Electronic Journal*, 2021. DOI: 10.2139/ssrn.3778525.

[131] S. Wibirama and K. Hamamoto, "Design and implementation of gaze tracking headgear for nvidia 3d vision®", in *2013 International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2013, pp. 84–87. DOI: 10.1109/ICITEED.2013.6676216.

[133] C. R. Cunha, A. Moreira, L. Pires, and P. O. Fernandes, "Intangible approaches to improve individual health indicators and empower caregivers", in *Internet of Everything for Smart City and Smart Healthcare Applications*, N. Gupta and S. Mishra, Eds. Cham: Springer Nature Switzerland, 2024, pp. 177–192, ISBN: 978-3-031-34601-9. DOI: 10.1007/978-3-031-34601-9_11.

[140] ——, "Using mixed reality and machine learning to assist caregivers in nursing home and promote well-being", *Procedia Computer Science*, vol. 219, pp. 1081–1088, 2023, ISSN: 1877-0509. DOI: 10.1016/j.procs.2023.01.387.

Filmography

[22] *Introducing Apple Vision Pro*, en-GB, Duration: 09m 21s, Jun. 2023. [Online]. Available: https://www.youtube.com/watch?v=TX9qSaGXFyg.

[79] Microsoft, *Microsoft at mwc19 barcelona*, en-GB, Duration: 01h 05m 30s, Feb. 2019. [Online]. Available: https://www.youtube.com/watch?v=rOeAek-Ltoo.

[89] T. Browne, *Did i pick the right database???*, en-GB, Duration: 01h 07m 47s, Aug. 2022. [Online]. Available: https://www.youtube.com/watch?v=cC6HFd1zcbo.

[139] EdRoh, *Build a complete react admin dashboard app | react, material ui, data grid, light & dark mode*, en-GB, Duration: 04h 04m 24s, Oct. 2022. [Online]. Available: https://www.youtube.com/watch?v=wYpCWwD1oz0.

Webography

[6] D. Barnard, *History of VR - Timeline of Events and Tech Development*, en, 2019. [Online]. Available: https://virtualspeech.com/blog/history-of-vr.

[8] M. Das, *[Tutorial] Augmented Reality with Internet of Things (IoT)*, May 2021. [Online]. Available: https://www.hackster.io/mithun-das/tutorial-augmented-reality-with-internet-of-things-iot-fb51e7.

[9] Microsoft, *TEI for Mixed Reality Partners | Microsoft HoloLens*. [Online]. Available: https://www.microsoft.com/en-us/hololens/forrester-total-economic-impact-isv.

[10] T. Metcalfe, *What is VR? The devices and apps that turn the real world virtual*, Mar. 2018. [Online]. Available: https://www.nbcnews.com/mach/science/what-vr-devices-apps-turn-real-world-virtual-ncna857001.

[11] D. Wong, *Research guides: Virtual Reality in the Classroom: What is VR?*, en, Jan. 2017. [Online]. Available: https://guides.library.utoronto.ca/c.php?g=607624&p=4938314.

[12] Adminsvr, *3dof vs 6dof in vr: What is the difference?*, Aug. 2022. [Online]. Available: https://www.smartvrlab.nl/3dof-vs-6dof-in-vr/.

[18] J. Coon, *Assisted Reality: What it Is, and How it Will Improve Service Productivity | PTC*, en, May 2018. [Online]. Available: https://www.ptc.com/en/blogs/service/what-is-assisted-reality.

[21] A. Rekdal, *What is spatial computing?*, Jun. 2023. [Online]. Available: https://builtin.com/hardware/spatial-computing.

[23] [Online]. Available: https://www.immersed.com/.

[24] *Openxr - high-performance access to ar and vr —collectively known as xr— platforms and devices*, Dec. 2016. [Online]. Available: https://www.khronos.org/openxr/.

[26] C. Bello, *Colombia makes history by hosting court hearing in the metaverse*, Mar. 2023. [Online]. Available: https://www.euronews.com/next/2023/03/01/future-of-justice-colombia-makes-history-by-hosting-its-first-ever-court-hearing-in-the-me.

[34] Vishnupriya, *Valley health system to build new smart hospital in new jersey*, Mar. 2023. [Online]. Available: https://www.hospitalmanagement.net/news/valley-health-system-new-smart-hospital/.

[35] *Investing in canada's first smart hospital: Cortellucci vaughan hospital*, Jan. 2021. [Online]. Available: https://explorecvh.ca/wp-content/uploads/2021/01/MHF_Smart-Hospital.20.final_.low_.pdf.

[37] G. Wiederrecht, S. Darwish, and A. Callaway, *The healthcare data explosion*. [Online]. Available: https://www.rbccm.com/en/gib/healthcare/episode/the_healthcare_data_explosion.

[39] *Patient details in the recycling? hospitals should cut down on paper to protect privacy: Study*, Mar. 2018. [Online]. Available: https://www.cbc.ca/news/canada/toronto/hospitals-documents-paper-patient-privacy-1.4584093.

[41] *Medical Record Management System: Making A Switch To Digital*, en, Mar. 2023. [Online]. Available: https://www.enter.health/post/medical-record-management-system-making-a-switch-to-digital.

[43] *What are the Advantages of Web-Based VS. Desktop-Based Inspection Software? | Fire Inspection Software*, en-US, Nov. 2018. [Online]. Available: https://www.formlinksystems.com/what-are-the-advantages-of-web-based-vs-desktop-based-inspection-software/.

[44] J. Russel, *Electronic health records*, en, May 2020. [Online]. Available: https://www.csp.org.uk/professional-clinical/digital-physiotherapy/electronic-health-records.

[49] *What is hl7 fhir?* [Online]. Available: https://www.tibco.com/reference-center/what-is-hl7-fhir.

[50] S. Sutner, *What is HL7 (Health Level Seven International)?*, en, Jun. 2015. [Online]. Available: https://www.techtarget.com/searchhealthit/definition/Health-Level-7-International-HL7.

[54] R. Chase, *Musk attorneys say Twitter stalling on document production*, Jul. 2022. [Online]. Available: https://dayton247now.com/news/nation-world/musk-attorneys-say-twitter-stalling-on-document-production-embattled-tesla-ceo-44-billion-bid-social-media-company-delaware-chancery-court-judge-proposed-case-schedule.

[80] *What is a kanban board and how to use it? basics explained.* [Online]. Available: https://kanbanize.com/kanban-resources/getting-started/what-is-kanban-board.

[81] V. Lo, *Sql vs nosql: Choosing database design*, Jan. 2021. [Online]. Available: https://lo-victoria.com/sql-vs-nosql-choosing-database-design.

[86] *NoSQL Databases List by Hosting Data - Updated 2023*, en-US. [Online]. Available: https://hostingdata.co.uk/nosql-database/.

[91] *How industry 4.0 technologies are changing manufacturing*. [Online]. Available: https://www.ibm.com/topics/industry-4-0.

[96] *Nb-iot and lte-m issues to boost lora and sigfox near and long-term lead in lpwa network connections*, Jun. 2019. [Online]. Available: https://www.abiresearch.com/press/nb-iot-and-lte-m-issues-boost-lora-and-sigfox-near-and-long-term-lead-lpwa-network-connections/.

[99] T. T. Network, *The things network*. [Online]. Available: https://www.thethingsnetwork.org/.

[101] T. G. Nagy, *Best HTTP/2 | Network | Unity Asset Store*. [Online]. Available: https://assetstore.unity.com/packages/tools/network/best-http-2-155981.

[102] E. Software, *OpenCV for Unity | Integration | Unity Asset Store*. [Online]. Available: https://assetstore.unity.com/packages/tools/integration/opencv-for-unity-21088.

[103] *Material Design*, https://m3.material.io/.

[104] P. Abeles, *Boofcv*, 2011-2021. [Online]. Available: http://boofcv.org/.

[105] P. Kumar, *Implementing async features in python - a step-by-step guide*. [Online]. Available: https://www.velotio.com/engineering-blog/async-features-in-python.

[108] Namit, *Hololens – continuous integration*, Nov. 2017. [Online]. Available: https://blog.kloud.com.au/2017/11/28/hololens-continuous-integration/.

[109] U. Technologies, *How il2cpp works*. [Online]. Available: https://docs.unity3d.com/560/Documentation/Manual/IL2CPP-HowItWorks.html.

[110] *Arcore supported devices*. [Online]. Available: https://developers.google.com/ar/devices.

[111] *Sranipal getting started steps*. [Online]. Available: https://forum.htc.com/topic/5642-sranipal-getting-started-steps/page/3/.

[120] N. C. Institute, *GraphQL Examples - GDC Docs*. [Online]. Available: https://docs.gdc.cancer.gov/API/Users_Guide/GraphQL_Examples/.

[121] *O que é um api gateway?* [Online]. Available: https://www.tibco.com/pt-br/reference-center/what-is-an-api-gateway.

[124] *The seven universal facial expressions of emotion*. [Online]. Available: https://leb.fbi.gov/image-repository/truth_8.jpg/view.

[125] A. Tandon, *GitHub - Tandon-A/emotic: PyTorch implementation of Emotic CNN methodology to recognize emotions in images using context information.* [Online]. Available: https://github.com/Tandon-A/emotic.

[128] *Unity job system and burst compiler: Getting started*. [Online]. Available: https://www.kodeco.com/7880445-unity-job-system-and-burst-compiler-getting-started.

[129] Valem, *GitHub - ValemVR/VR-Game-Jam-Template: Starter Project using Unity XR Toolkit created for VR Jam 2023*, Jun. 2023. [Online]. Available: https://github.com/ValemVR/VR-Game-Jam-Template.

[130] karboosx, *Sci-Fi Styled Modular Pack | 3D Sci-Fi | Unity Asset Store*. [Online]. Available: https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-styled-modular-pack-82913.

[132] Shapes, *Nature Starter Kit 2 | 3D Environments | Unity Asset Store*. [Online]. Available: https://assetstore.unity.com/packages/3d/environments/nature-starter-kit-2-52977.

[134] A. Vidal, *Emotion Icons For Presentations (free download)*, 2018. [Online]. Available: https://www.linkedin.com/pulse/emotion-icons-presentations-free-download-anna-vital.

[135] https://www.blueraja.com/blog/author/blueraja, *Using Vector3.Lerp() correctly in Unity*. [Online]. Available: https://www.blueraja.com/blog/404/how-to-use-unity-3ds-linear-interpolation-vector3-lerp-correctly.

[136] *Free Photo | Smiley man posing indoors front view*. [Online]. Available: https://www.freepik.com/free-photo/smiley-man-posing-indoors-front-view_33803051.htm#from_view=detail_serie.

[137] M. Corbett, *Github - mattycorbett/hololens2-facial-detection-and-localization: Project to port existing ms holographic face tracking sample to unity and c#*. [Online]. Available: https://github.com/mattycorbett/HoloLens2-Facial-Detection-and-Localization.

[138] circuscompany, *Clean SCI-FI GUI Pack | 2D GUI | Unity Asset Store*. [Online]. Available: https://assetstore.unity.com/packages/2d/gui/clean-sci-fi-gui-pack-183901.

Appendix A Original Thesis Proposal

INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

MESTRADO EM INFORMÁTICA

Master of Informatics

Unidade Curricular de "Dissertação/Projeto/Estágio"
Course unit "Thesis/Project/Intership"

Work proposal

Proposta de tema

☑ Dissertação *Thesis* ☐ Projeto *Project* ☐ Estágio *Internship*

Título *Title*

The use of Extended Reality and Machine Learning to Improve Healthcare and Promote GreenHealth

Palavras-chave *Keywords*

Extended Reality; Machine Learning; GreenHealth; Elderly;

Orientador IPB *IPB Supervisor*	*email*
Carlos Rompante Cunha	crc@ipb.pt

Co-orientador IPB *IPB Co-supervisor*	*email*
Rui Pedro Lopes	rlopes@ipb.pt

Co-orientador externo 1 *External co-supervisor 1*	*email*

Instituição do Co-orientador externo 1 *Institution of external co-supervisor 1*	País *Country*

Co-Orientador externo 2 *External co-supervisor 2*	*email*

Instituição do Co-orientador externo 2 *Institution of external co-supervisor 2*	*Country*

Aluno *Student*

n º.	Nome *name*	email *email*
38978	**André Silva Moreira**	a38978@alunos.ipb.pt

Objetivos *Goals*

- Review of the state-of-art related to the technology used in health care and by caregivers;
- Study about the potential of Extended Reality (XR) and Machine Learning (ML) in the context of greenhealth and healthcare;
- Propose of a conceptual model, with working prototypes for validation.

Descrição *Description*

This work will:

Review of the state-of-art related to the use of technology in the health and caregivers services with focus on elders. Trying to understand what currently is used and with which results.

Study of the potential of combining XR and ML to develop new approaches to promote wellbeing among elderly people and also to support caregivers daily routines.

Propose a integrated conceptual model to promote wellbeing and support caregivers daily routines in the context of nursing homes.

Prototype some aspects of the proposed model.

Metodologia / Plano de trabalhos *Methodology/ Work plan*

- Reading and analysis of bibliographic references, either to study the scope of application, or to analyze technologies that are being used and the results that have been obtained;

- Understanding how XR combined with ML can help to develop new models to improve the well-being of the pacients, especially in the context of nursing homes, and support caregivers;

- Design of a conceptual model (according to the objectives already stated);

- Validation of some aspects of the model through prototyping and tests in a real scenario.

Pré-requisitos *Prerequisites*

Recursos necessários *Resources needed*

*Desktop; Virtual Reality Headset with Eye Tracking (pref: HTC Vive Pro Eye), Mixed Reality Headset (pref: HoloLens 2);
(*) existing equipment in EsACT - IPB.

Data *Date*

09/11/2022

Appendix B Publications

B.1 Citations

- Handle (IPB): 10198/28698

  Cunha, Carlos R.; Moreira, André Silva; Pires, Luís; Fernandes, Paula O. (2024). Intangible approaches to improve individual health indicators and empower caregivers. In Internet of everything for smart city and smart healthcare applications. Switzerland: Springer. ISBN 978-3-031-34600-2

- Handle (IPB): 10198/28549

  Cunha, Carlos R.; Moreira, André Silva; Pires, Luís; Fernandes, Paulo O. (2023). Using mixed reality and machine learning to assist caregivers in nursing home and promote well-being. In International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies - Centeris 2022. Procedia Computer Science. p. 1081-1088. ISSN 1877-0509

Appendix C Nginx Configuration (Load Balancing / Reverse-Proxies / WS / SSE)

Configuration file of the Nginx running on the Cloud Server, which contains the Web API, Web-sockets, Server-Sent Events and makes requests to the On-Premise Server.

```
log_format upstreamlog '$server_name to: $upstream_addr {$request} '
   'upstream_response_time $upstream_response_time'
   ' request_time $request_time';

upstream qrAuth  {
        server 127.0.0.1:9010;
}
upstream mlFaceRecon {
        server 127.0.0.1:9000;
}
upstream qrDecode {
        server 127.0.0.1:9020;
}
upstream vrHealSession {
        server 127.0.0.1:9030;
        server 127.0.0.1:9031;
}
upstream vrHealSessionPanoramicStream {
        server 127.0.0.1:9040;
        server 127.0.0.1:9041;
}
```

```
upstream sse {
        server 127.0.0.1:9060;
}

server {
        listen 80 default_server;
        listen [::]:80 default_server;
        access_log /var/log/nginx/nginx-access.log upstreamlog;
        root /var/www/html;
        index index.html index.htm index.nginx-debian.html;

        location /nginx {
                try_files $uri $uri/ =404;
        }
        location /api {
                proxy_pass http://localhost:8000
        }
        location /authChannel {
                proxy_pass http://localhost:8000;
        }
        location /ws/ml/emotionFaceRecon {
                proxy_pass http://mlFaceRecon;
                include proxy_params;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                proxy_set_header Host $http_host;
        }

        < Others Websockets >

        location /sse/vr/heal/session/panoramic/stream/receiver {
                proxy_pass http://sse;
                proxy_buffering off;
                proxy_cache off;
```

```
57                  proxy_set_header Host $http_host;
58                  proxy_set_header Connection '';
59                  proxy_http_version 1.1;
60                  chunked_transfer_encoding off;
61          }
62  }
```

Appendix D Code Samples

D.1 Realm Object-Oriented Database Management

Class *UserEntity*, definition of the data as to be saved.

```
using Realms;
using System.Collections.Generic;

class UserEntity : RealmObject {
    [PrimaryKey]
    public string Email { set; get; }

    public string UUID { set; get; }
    public string CurrentRole { set; get; }
    public string Token { set; get; }
    public IList<MemberOf> MemberOf { get; }
    public UserEntity() { }
    public UserEntity(string email, string UUID, string token) {
        this.Email = email;
        this.UUID = UUID;
        this.Token = token;
    }
}
```

Code Sample with Imports, Realm Configuration, Database Reset and User logged CRUD

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Realms;
using System;
using Newtonsoft.Json.Linq;
using System.Runtime.CompilerServices;
using System.Linq;
using Debug = XRDebug;

public static class RealmManager {
    private static RealmConfiguration realmConfig = new RealmConfiguration
    {
        SchemaVersion = 1
    };
    public static Realm realm {
        get {
            realmConfig.ShouldDeleteIfMigrationNeeded = true;
            return Realm.GetInstance(realmConfig);
        }
        private set {
            realm = value;
        }
    }

    public static void BulldozeRealm() {
        using (var realm = RealmManager.realm) {
            realm.Write(() => {
                realm.RemoveAll();
            });
        }
    }

```

```
33      public static string FindActiveUser(bool save = false) {
34          List<UserEntity> userObject = RealmManager.realm.All<UserEntity>()
    .Where(user => user.Token != "").ToList();
35          if (userObject.Count > 0 && save) {
36              AccountManager.Token = userObject[0].Token;
37              AccountManager.ActiveUserEmail = userObject[0].Email;
38              return userObject[0].Token;
39          }
40          return null;
41      }
42
43      public static bool LogoutUser(string userEmail) {
44          RealmObject userObject = RealmManager.realm.Find<UserEntity>(
    userEmail);
45          using (Realm realm = RealmManager.realm) {
46              using (Transaction transaction = realm.BeginWrite()) {
47                  try {
48                      (userObject as UserEntity).Token = "";
49                      realm.Add(userObject, update: true);
50                      transaction.Commit();
51                      return true;
52                  } catch (Exception ex) {
53                      transaction.Rollback();
54                      return false;
55                  }
56              }
57          }
58      }
59
60      public static bool CreateUpdateUser(string uuid, string token, string
    userEmail) {
61          RealmObject userObject = RealmManager.realm.Find<UserEntity>(
    userEmail);
62          using (Realm realm = RealmManager.realm) {
63              using (Transaction transaction = realm.BeginWrite()) {
```

```
                try {
                    if (userObject == null) {
                        userObject = new UserEntity(
                                email: userEmail,
                                UUID: uuid,
                                token: token
                        );
                        realm.Add(userObject);
                    } else {
                        (userObject as UserEntity).Token = token;
                        realm.Add(userObject, update: true);
                    }
                    transaction.Commit();
                    return true;
                } catch (Exception ex) {
                    transaction.Rollback();
                    return false;
                }
            }
        }
    }
```

D.2 MongoDB Query to calculate Cosine Similarity for Face Recognition

Requires a string to `collection`, being referred to the DB collection containing the embeddings, while targets is the array of embeddings from the image to be recognised.

```
async def QueryDatabase(self, collection, targets):
    return self.dbClient[self.database][collection].aggregate( [ {
        "$addFields": {
                "target_embeddings": targets
        } }, {
                "$unwind": "$embeddings"
            }, {
                "$unwind": "$target_embeddings"
            }, {
                "$addFields": {
                    "dot_product": {
                        "$reduce": {
                            "input": {
                                "$zip": {
                                    "inputs": ["$embeddings.embedding", "
$target_embeddings.embedding"] }
                            },
                            "initialValue": 0,
                            "in": {
                                "$add": [
                                    "$$value", {
                                        "$multiply": [
                                            { "$arrayElemAt": ["$$this",
0] },
                                            { "$arrayElemAt": ["$$this",
1] }
                                        ]
```

```
                                    }
                                ]
                            }
                        }
                    },
                    "magnitude_product": {
                        "$sqrt": {
                            "$multiply": [
                                { "$sqrt": { "$sum": { "$map": { "input": "$embeddings.embedding", "as": "e", "in": { "$pow": ["$$e", 2] } } } } },
                                { "$sqrt": { "$sum": { "$map": { "input": "$target_embeddings.embedding", "as": "t", "in": { "$pow": ["$$t", 2] } } } } }
                            ] } } }
            },
            {
                "$addFields": {
                    "similarity": {
                        "$divide": ["$dot_product", "$magnitude_product"]
                    } }
            }, {
                "$match": {
                    "similarity": { "$gte": 0.75 }
                }
            }, {
                "$sort": {
                    "similarity": -1
                }
            }, {
                "$group": {
                    "_id": {
                        "uuid": "$uuid",
                        "target_id": "$target_embeddings.id" },
                    "max_similarity": { "$first": "$similarity" },
```

```
                    "count": { "$sum": 1 }
                } },
            {
                "$match": {
                    "count": { "$gt": 2 }
                }
            }, {
                "$group": {
                    "_id": None,
                    "results": {
                        "$push": {
                            "uuid": "$_id.uuid",
                            "target_id": "$_id.target_id"
                        }
                    }
                }
            }, {
                "$unwind": "$results"
            }, {
                "$replaceRoot": {
                    "newRoot": "$results"
                }
            }, {
                "$project": {
                    "_id": 0
                }
            } ] )
```

D.3 ScriptableObject for Interface Management

```
[CreateAssetMenu(menuName = "Scriptable Object/Graphic User Interface")]
public class GraphicUserInterfaceScriptableObject : ScriptableObject {
    [System.Serializable]
    public class data {
        public WindowType windowType;
        public GameObject window;
        public List<windowComponents> components = new List<windowComponents>();
        public Dictionary<string, string> componentsDict = new Dictionary<string, string>();
    }
    [System.Serializable]
    public struct windowComponents {
        public string name;
        public GUIComponentType type;
        public string path;
    }
    [SerializeField] data[] _windows;
    public data[] windows {
        get {
            return _windows;
        }
    }
    public void SetupComponentsDictionary() {
        foreach (var i in windows) {
            foreach (var n in i.components) {
                i.componentsDict.Add(n.name, n.path);
            }
        }
    }
}
```

D.4 GraphQL Schema and Request Body for Login

A GraphQL Schema is language-agnostic, having is own syntax and semantic, some important keywords are `type` to define what can be comparable as structures of data, as `Query` and `Mutation`. However there are a variety of keywords that empower the developer in simplifying the schema, such as `union` which enables a request to return different structures as answer.

```
interface BaseIdentification {
    uuid: String
    label: String
}
type Member implements BaseIdentification {
    uuid: String
    label: String
    name: String
    password: String
    token: String
    username: String
    email: String
    memberOf: [MemberOf]
}
type MemberOf {
    role: String
    institution: Institution
}
type Institution implements BaseIdentification {
    uuid: String
    label: String
    name: String
}
type Error {
  message: String!
  description: String
```

```
}

input LoginCredentials {
  email: String!
  password: String!
}

union MemberLoginResponse = Member | Error

type Query {
  MemberLogin(loginCredentials: LoginCredentials): MemberLoginResponse
}
```

A particularity of GraphQL is that only uses POST requests and one route, which means that the body of the request is where will be defined the request.

```
query ($email: String!, $password: String!) {
    MemberLogin (loginCredentials: { email: $email, password: $password }) {
        ... on Member { uuid, name, username, email, token, memberOf { role, institution { uuid, name } }
        } ... on Error { message }
    }
}
```

Some of the request may require to sent data to the server, like filters, depending on the programming language or library this may be needed to be inserted directly in the query or aside, in the case of Postman, there is the possibility of defining `GraphQL Variables` with the following structure.

```
{ "email":"caregiver@carexr.com",
    "password": "password" }
```

D.5 Building GraphQL Queries in Unity

Unity is a powerful game engine and able to execute web requests, however is still not ready to the use, in part due to the small niche of applications using both technologies. The Unity Asset used to simplify and optimize the use of web requests, Best HTTP/2, also does not support the GraphQL queries and while others assets were found to make use of GraphQL, these or are only able to process basic queries or are outdated.

It shall be noted that both methods are able to do POST requests, the problem resides in mounting the body content without syntax errors, since while similar to a JSON, it is not. As such was necessary to developer a solution for this particular problem, starting by defining a set of structures.

```
1  public static class GraphQL {
2      public struct Type {
3          public string name;
4          public Params[] parameters;
5          public Type[] subfield;
6
7          public Type(string name, Params[] parameters = null) {
8              this.name = name;
9              this.parameters = parameters;
10             this.subfield = null;
11         }
12
13         public Type(string name, Type[] subfield) {
14             this.name = name;
15             this.parameters = null;
16             this.subfield = subfield;
17         }
18
19         public Type(string name, Params[] parameters, Type[] subfield) {
20             this.name = name;
```

```
            this.parameters = parameters;
            this.subfield = subfield;
        }
    }

    public struct Params {
        public string name;
        public string value;

        public Params(string name, string value) {
            this.name = name;
            this.value = value;
        }
    }
}
```

The developed structures are used to simply the process of mounting query by only needing to define hierarchy and the data required.

```
GraphQL.Type queryOperation = new GraphQL.Type(
    "medicationToTake", new GraphQL.Params[] {
        new GraphQL.Params("memberID", "\"" + AccountManager.
    ActiveUserEmail + "\""),
                    new GraphQL.Params("institutionID", "\"" +
    institutionUUID + "\""),
    }
);

await APIManager.ExecuteRequest(RealmManager.realm.Find<UserEntity (
    AccountManager.ActiveUserEmail).Token.ToString().Trim(), queryOperation
    ,
            //On succeed code
            ,
            new GraphQL.Type[] {
```

```
                new GraphQL.Type("atTime"),
                new GraphQL.Type("quantity"),
                new GraphQL.Type("timeMeasure"),
                new GraphQL.Type("intOfTime"),
                new GraphQL.Type("medication", new GraphQL.Type[] {
                    new GraphQL.Type("uuid"),
                    new GraphQL.Type("Name")
                }),
                new GraphQL.Type("pacient", new GraphQL.Type[] {
                    new GraphQL.Type("uuid")
                })
            }
        );
```

While a generic method is able to mount the query.

```
private static void MountQuery(GraphQL.Type[] args, ref string query, byte identationLevel = 2) {
    foreach (GraphQL.Type field in args) {
        query += (new string('\t', identationLevel) + field.name);
        if (field.parameters != null) {
            query += " (";
            for (byte index = 0; index < field.parameters.Length; index++)
                query += (field.parameters[index].name + ": " + field.parameters[index].value + (index >= field.parameters.Length ? ", " : ""));
            query += ") {";
        }
        if (field.subfield != null) {
            query += " {\r\n";
            MountQuery(field.subfield, ref query, identationLevel += 1);

            query += (new string('\t', identationLevel - 1) + "}\r\n");
        } else
```

```
16              query += "\r\n";
17      }
18  }
19
20  public static async Task ExecuteRequest(string token, GraphQL.Type type,
        Action<string, bool> action, params GraphQL.Type[] args) {
21      await Task.Run(() => {
22          string query = ""; //query {\r\n
23          query += (new string('\t', 1) + type.name);
24          if (type.parameters != null) {
25              query += " ( ";
26              foreach (GraphQL.Params parameter in type.parameters)
27                  query += (parameter.name + ": " + parameter.value + ", ");
28              if (type.subfield != null)
29                  query += " ) {\r\n";
30          }
31          if (type.subfield != null) {
32              query += " ( ";
33              foreach (GraphQL.Type subfield in type.subfield) {
34                  query += subfield.name + ": { ";
35                  if (subfield.parameters != null) {
36                      foreach (GraphQL.Params parameter in subfield.
        parameters)
37                          query += (parameter.name + ": " + parameter.value
        + ", ");
38                  }
39
40                  query += " } ";
41              }
42              query += " ) { \r\n";
43          } else
44              query += " ) { \r\n";
45
46          MountQuery(args, ref query, 2);
47          query += (new string('\t', 1) + "}\r\n");
```

```
48          query = "query { " + query + "}";
49
50          query = query.Replace("\n", " ").Replace("\t", " ").Replace("\r",
    " ");
51          RegexOptions options = RegexOptions.None;
52          Regex regex = new Regex(@"[ ]{2,}", options);
53          query = regex.Replace(query, @" ");
54
55          string jsonData = JsonConvert.SerializeObject(new { query });
56          byte[] postData = Encoding.ASCII.GetBytes(jsonData);
57
58          using (HTTPRequest request = new HTTPRequest(new Uri(_httpProtocol
     + _ip + _port + GraphqlPath), HTTPMethods.Post, (HTTPRequest request,
    HTTPResponse response) => OnRequestFinished(action, request, response))
    ) {
59              request.DisableCache = true;
60              request.SetHeader("Content-Type", "application/json");
61              request.SetHeader("Accept", "application/json");
62              request.SetHeader("Keep-Alive", "timeout = 2, max = 20");
63
64              if (token != "")
65                  request.SetHeader("Authorization", token.Trim());
66
67              request.RawData = Encoding.UTF8.GetBytes(jsonData);
68              request.Send();
69          }
70      });
71  }
```

D.6 gRPC Proto file for Emotion Recognition Microservice

```
syntax = "proto3";
package emotionRecognition;

service EmotionRecognitionService {
    rpc Inference (EmotionRecognitionRequest) returns (
   EmotionRecognitionInferenceReply) { }
}

message EmotionRecognitionRequest {
    bytes image = 1;
    map<string, int32> personBox = 2;
}

// "repeated" means list of
message EmotionRecognitionInferenceReply {
    map<string,float> continuous = 1;
    repeated string categorical = 2;
}
```

D.7 Applying Panoramic Image to Sphere in runtime and bounding box's setup

The following code is responsible to apply the Panoramic Image to a sphere, being the Panoramic Image applied to a `Textute2D` previously, by default the texture is applied to the "outside" surface of the sphere which is not the goal, as such is necessary to invert the `mesh` normals, which normally points outward, perpendicular to the mesh surface, there's *N* normals, each associated to a vertex.

```
public static void ApplySphereTexture(ref GameObject sphere, Texture2D
    panoramicTexture) {
     _currentSphere = sphere;
     MeshFilter meshFilter = sphere.GetComponent<MeshFilter>();
     if (meshFilter != null) {
          Mesh mesh = meshFilter.mesh;
          Vector3[] normals = mesh.normals;
          for (int i = 0; i < normals.Length; i++)
              normals[i] = -normals[i];
          mesh.normals = normals;
          int[] triangles = mesh.triangles;
          for (int i = 0; i < triangles.Length; i += 3) {
              int temp = triangles[i];
              triangles[i] = triangles[i + 1];
              triangles[i + 1] = temp;
          }
          mesh.triangles = triangles;
     }
     Material sphereMaterial = new Material(Shader.Find("Standard"));
     sphereMaterial.mainTexture = panoramicTexture;
     Renderer sphereRenderer = sphere.GetComponent<Renderer>();
     if (sphereRenderer != null)
          sphereRenderer.sharedMaterial = sphereMaterial;
}
```

The method ConvertBoundingBoxCenterTo3D is used to get the center of a bounding box and get is respective position in the surface of the sphere, being used inside of MountHotspots.

```
private static Vector3 ConvertBoundingBoxCenterTo3D(JToken boundingBoxData
    ) {
     float centerX = (float)boundingBoxData["x"] + (float)boundingBoxData["
    width"] / 2f;
     float centerY = (float)boundingBoxData["y"] + (float)boundingBoxData["
    height"] / 2f;
     float theta = (centerY / panoramicImageHeight) * Mathf.PI;
     float phi = (centerX / panoramicImageWidth) * Mathf.PI * 2f;
     float radius = sphereDiameter / 2f;
     Vector3 spherePosition = new Vector3(
         radius * Mathf.Sin(theta) * Mathf.Cos(phi),
         radius * Mathf.Cos(theta),
         radius * Mathf.Sin(theta) * Mathf.Sin(phi)
     );
     return spherePosition;
}
```

ApplySphereTexture only creates the "scenario", while may be enough for certain use cases, for the the Eye Tracking exercise is necessary GameObjects with a Collider and the necessary scripts from Tobii XR to be able to detect the gaze behaviour.

Those GameObjects are the equivalent to the Bounding Box's predefined in another application and saved in the DB, however those bounding box's were defined in a 2D image, being necessary to convert both their position as well the scale to 3D.

```
public static void MountHotspots(JToken data, Action onComplete) {
     panoramicImageWidth = (int)data["imageWidth"];
     panoramicImageHeight = (int)data["imageHeight"];
     _hotspotContainer = GameObject.CreatePrimitive(PrimitiveType.Sphere);
     _hotspotContainer.transform.position = Vector3.zero;
     _hotspotContainer.transform.rotation = Quaternion.identity;
```

```
7       _hotspotContainer.transform.localScale = Vector3.one * sphereDiameter;
8       _hotspotContainer.gameObject.name = "Hotspot Container";
9       float radius = sphereDiameter / 2f;
10      float theta, phi, x, y;
11      foreach (JToken hotspot in data["mapping"]) {
12          if (hotspot["boundingBox"].HasValues) {
13              Vector3 centerPosition = ConvertBoundingBoxCenterTo3D(hotspot
    ["boundingBox"]);
14              GameObject boundigBox = GameObject.CreatePrimitive(
    PrimitiveType.Cube);
15              x = (float)hotspot["boundingBox"]["x"];
16              y = (float)hotspot["boundingBox"]["y"];
17              theta = (y / panoramicImageHeight) * Mathf.PI;
18              phi = (x / panoramicImageWidth) * Mathf.PI * 2f;
19              Vector3 x1y1 = new Vector3(
20                  radius * Mathf.Sin(theta) * Mathf.Cos(phi),
21                  radius * Mathf.Cos(theta),
22                  radius * Mathf.Sin(theta) * Mathf.Sin(phi)
23              );
24              x = (float)hotspot["boundingBox"]["x"] + (float)hotspot["
    boundingBox"]["width"];
25              y = (float)hotspot["boundingBox"]["y"];
26              theta = (y / panoramicImageHeight) * Mathf.PI;
27              phi = (x / panoramicImageWidth) * Mathf.PI * 2f;
28              Vector3 x2y1 = new Vector3(
29                  radius * Mathf.Sin(theta) * Mathf.Cos(phi),
30                  radius * Mathf.Cos(theta),
31                  radius * Mathf.Sin(theta) * Mathf.Sin(phi)
32              );
33              Vector3 width = x1y1 - x2y1;
34              x = (float)hotspot["boundingBox"]["x"] + (float)hotspot["
    boundingBox"]["width"];
35              y = (float)hotspot["boundingBox"]["y"] + (float)hotspot["
    boundingBox"]["height"];
36              theta = (y / panoramicImageHeight) * Mathf.PI;
```

```
            phi = (x / panoramicImageWidth) * Mathf.PI * 2f;
            Vector3 x2y2 = new Vector3(
                radius * Mathf.Sin(theta) * Mathf.Cos(phi),
                radius * Mathf.Cos(theta),
                radius * Mathf.Sin(theta) * Mathf.Sin(phi)
            );
            Vector3 height = x2y2 - x2y1;
            boundigBox.gameObject.transform.localScale = new Vector3(width.magnitude, height.magnitude, 0.0005f);
            boundigBox.gameObject.transform.position = centerPosition;
            boundigBox.gameObject.transform.LookAt(new Vector3(0, 0, 0));
            boundigBox.transform.parent = _hotspotContainer.transform;
            HotspotHandler hotspotHandler = boundigBox.AddComponent<HotspotHandler>();
            boundigBox.gameObject.layer = LayerMask.NameToLayer("Hotspot");
            boundigBox.gameObject.name = hotspot["data"]["alias"].ToString() + " [" + hotspot["uuid"].ToString() + "]";
            hotspotHandler.SetHotspotData(hotspot["data"]["alias"].ToString(), hotspot["uuid"].ToString(), hotspot["data"]["content"]);
            }
        }
        _hotspotContainer.transform.position = new Vector3(0, 1.43f, 0);
        onComplete?.Invoke();
        return;
    }
}
```

Both methods, `ApplySphereTexture` and `MountHotspots`, require a sphere in the scene, in the following code this sphere is created in runtime after being loaded the scene "Panoramic Session", after which the methods are invoked.

```
SceneTransitionManager.Instance.GoToSceneAsync(SceneTransitionManager.
    Scenes["Panoramic Session"], () => {
     GameObject sphere = GameObject.CreatePrimitive(PrimitiveType.Sphere);
     sphere.transform.position = new Vector3(0, 1.43f, 0);
     sphere.transform.localScale = new Vector3(9, 9, 9);
     sphere.transform.rotation = Quaternion.identity;
     sphere.transform.Rotate(Vector3.up, -180);
     sphere.gameObject.name = "360 Image";
     PanoramicManager.ApplySphereTexture(ref sphere, PanoramicManager.
    CurrentHotspotTexture);
     PanoramicManager.MountHotspots(executionRequest["params"], () => {
         < Code to be executed when finished mounting hotspots/bounding box
    's >
     });
     <...>
```

Appendix E Images

E.1 Web Application Screens

E.1.1 Login Screen

In the following image (Figure E.1) the user is asked to insert a email and password to do login, considering the context, no register step was putted in case. The query is done through the GraphQL request, being the user saved in the Neo4j DB, the email is used as unique identifier for login validation, being used Bcrypt to encrypt the password.

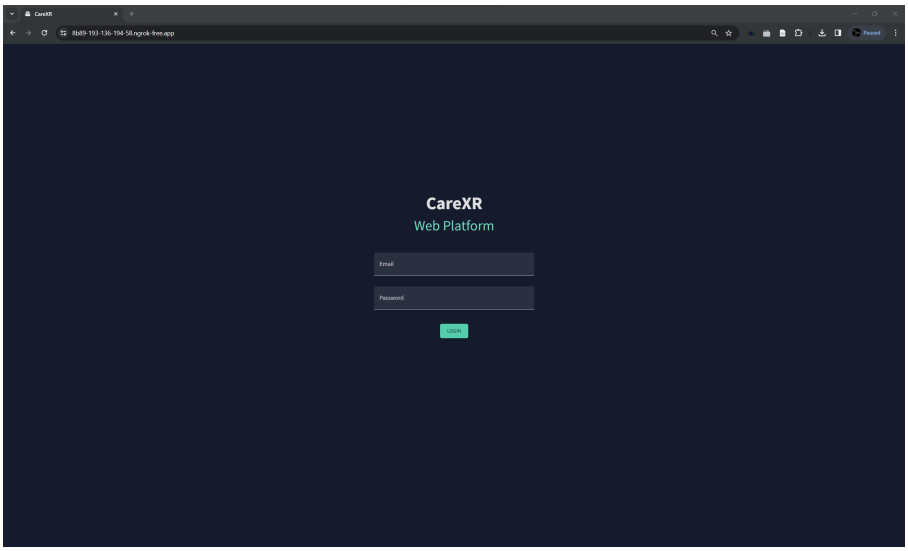

Figure E.1: Login Screen

E.1.2 VR Session Screens

In the initial screen of the VR Session (Figure E.2) the user is asked to insert the code of the session being that a NanoID displayed in the VR Application

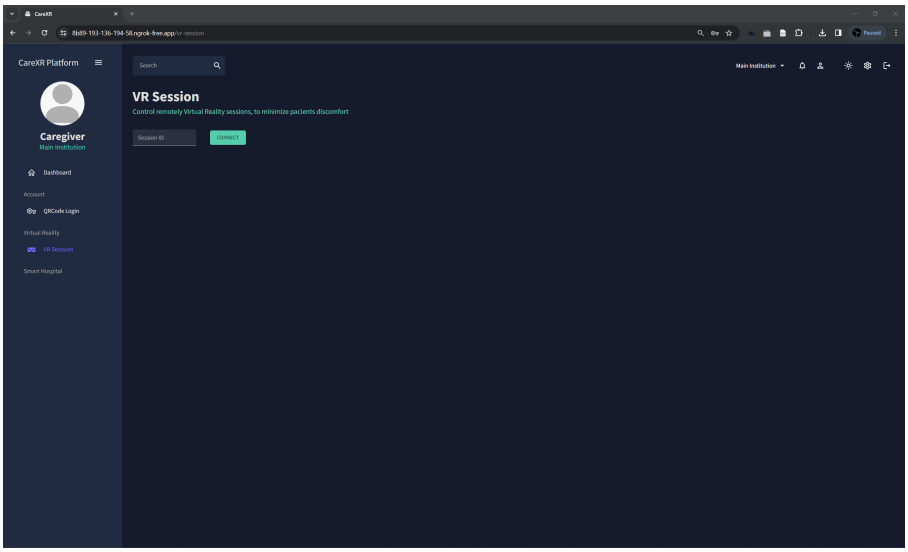

Figure E.2: VR Session: Start Screen

After being established the connection the user can choose which "scenario" to use for the exercise (Figure E.3).

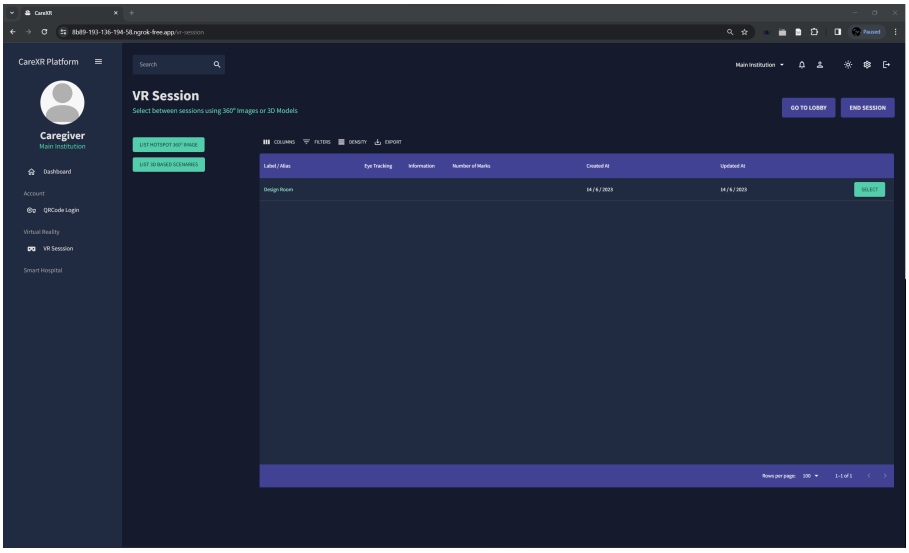

Figure E.3: VR Session: Exercise Scenario Choice

When chosen a Panoramic Image as scenario, it is displayed in the Web Application (Figure E.4), being also asked for definition of the exercise type and label.

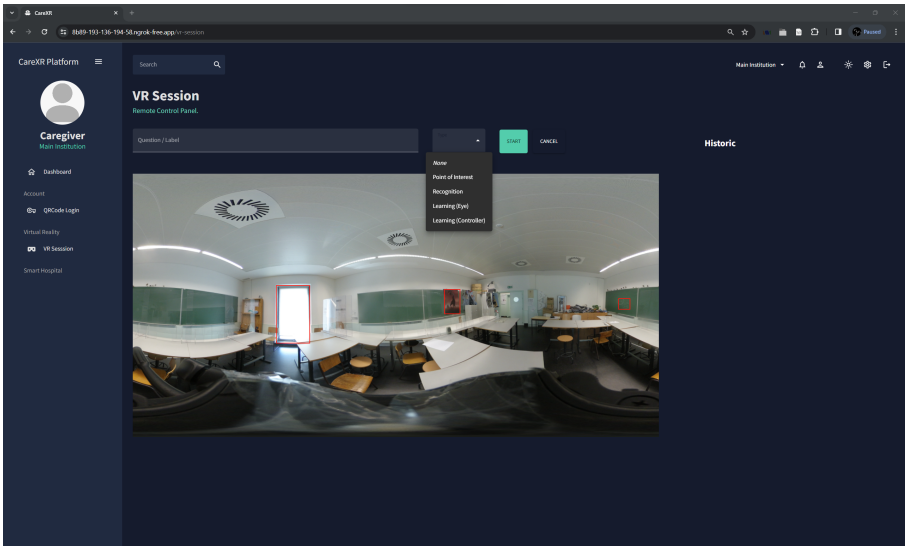

Figure E.4: VR Session: Exercise Type Choice

After filled the necessary information's the user can start the exercise, resulting in the screen shown in Figure E.5.

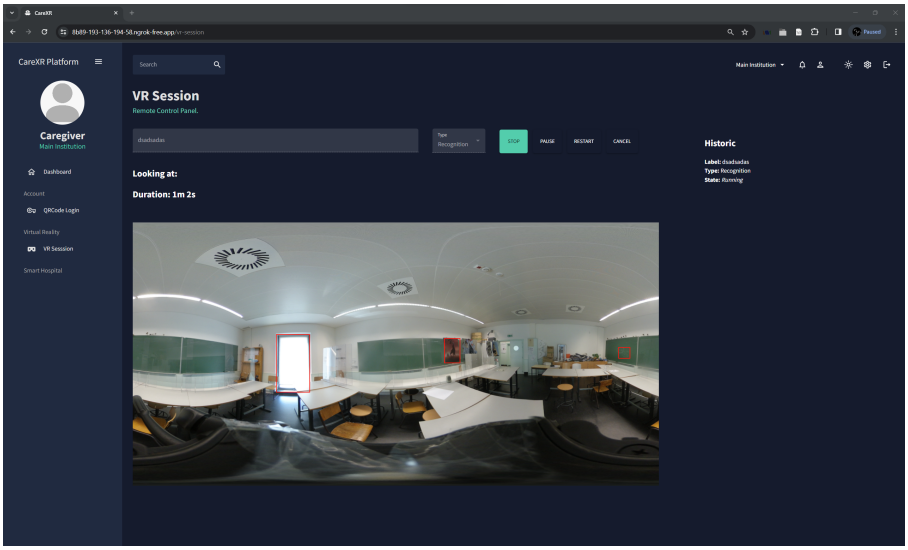

Figure E.5: VR Session: Exercise Running

The user can then end the exercise being displayed a log (Figure E.6), respective to the type of exercise.

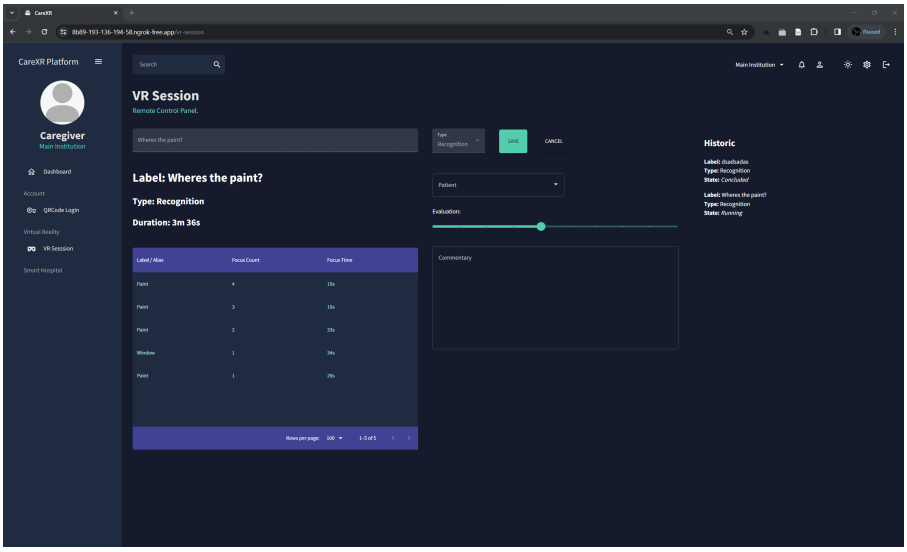

Figure E.6: VR Session: Exercise Results

E.1.3 QRCode Auth Screen

In the screen (Figure E.7) the user can request a QR Code to do authentication in another application able to capture pictures, in this project being the MR Application.

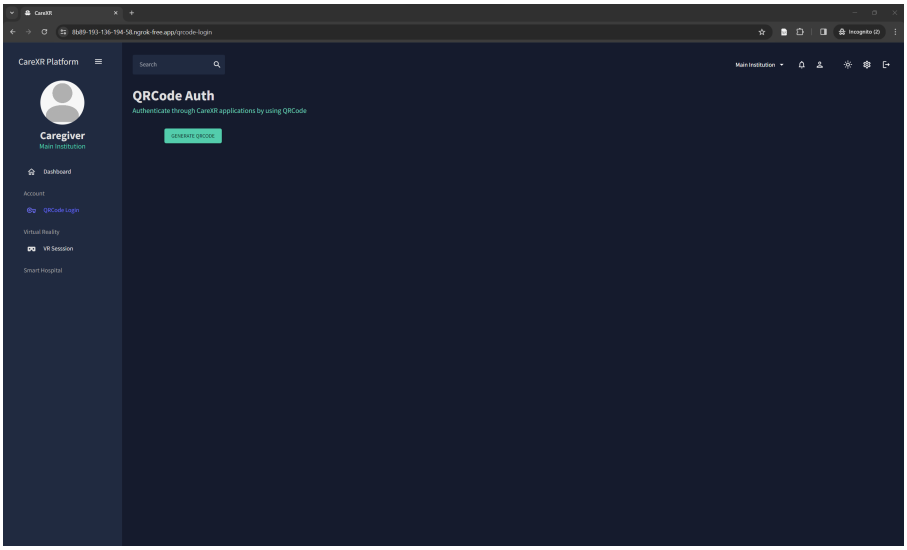

Figure E.7: QRCode Auth Screen

E.2 Mixed Reality Interfaces Samples in Unity Build

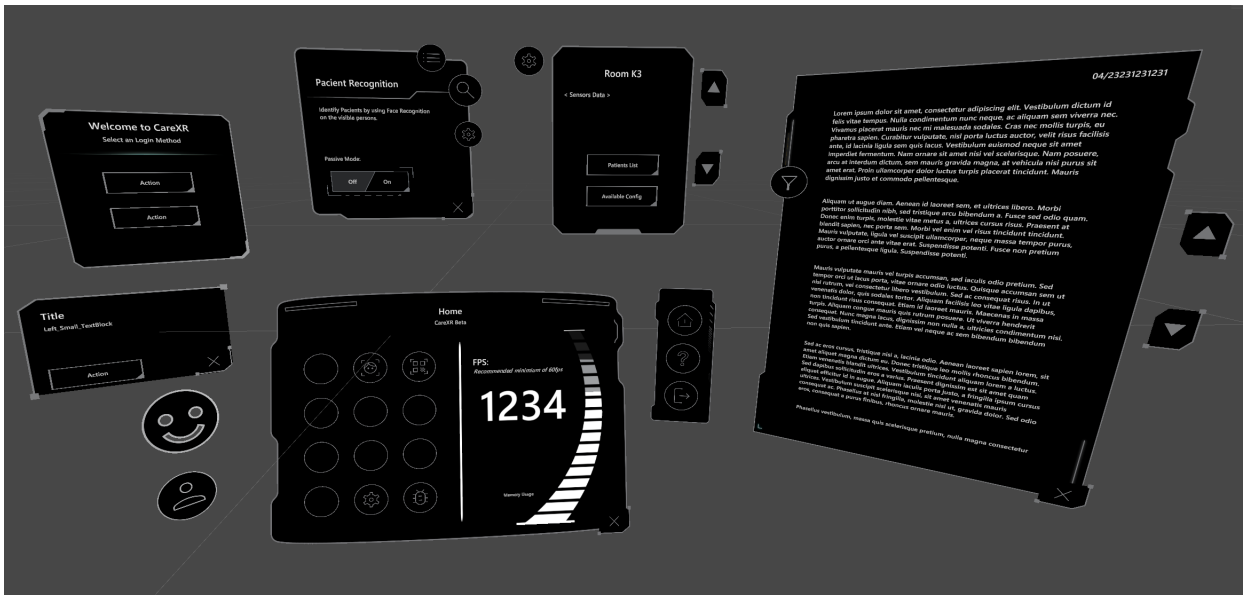

Figure E.8: Mixed Reality Interfaces Samples in Unity Build

Glossary

API An Application Programming Interface (API) is a particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another particular software program that implements that API.

Back-End The server-side of a software application responsible for handling data storage, processing, and managing the logic that enables front-end applications to function.

Base64 A binary-to-text encoding scheme commonly used to represent binary data, such as images or files, as ASCII characters. It's used for data transmission and storage in various applications.

Canvas In Unity, a Canvas is a crucial component for creating and managing 2D user interfaces (UI) within games. It serves as a platform for designing and displaying UI elements like buttons, text, and images, enhancing the user experience and interactivity of Unity applications.

Collider In game development and computer graphics, a collider is a component that defines the shape and physical interactions of an object within a game world, used for collision detection and response.

Cybersickness Discomfort or nausea experienced by individuals when exposed to virtual environments or immersive technologies, such as virtual reality or augmented reality, often due to a mismatch between visual stimuli and physical motion.

Dataset Is a structured collection of data points, organized for analysis or machine learning. It can take various forms such as spreadsheets, databases, or text/image files, providing a foundation for extracting insights and training models.

Diegetic Sound Audio elements that are an inherent part of the virtual environment and originate from within the simulated world. These sounds, such as footsteps or environmental noises, contribute to the immersive experience by aligning with the user's actions and surroundings in the virtual space. Unlike non-diegetic sound (e.g. select/click sound), these are integral to the virtual environment.

Framework A pre-defined structure or set of tools that provides a foundation for building software applications. Frameworks offer a set of rules, conventions, and components to streamline the development process. They often dictate the overall architecture and flow of an application.

Front-End The user interface and client-side components of a software application that users interact with directly. It is responsible for presenting data and enabling user interactions.

GameObject In the context of game development and game engines, a GameObject is a fundamental entity that represents an object or character in a game. It can contain components, behaviors, and properties as well scripts.

JWT A compact, self-contained method for securely transmitting information between parties as a JSON object. JWTs are often used for user authentication and authorization in web services.

Library A collection of pre-written code modules or functions that can be used to add specific features or capabilities to a software application. Libraries are designed to be reused and can help developers save time and effort by providing ready-made solutions to common programming tasks.

Material (in Unity) A scriptable asset in the Unity game engine that governs the visual appearance of surfaces within a 3D environment. Unity materials define properties such as color, texture, transparency, and shading, contributing to the overall visual presentation of objects in a Unity scene.

Network Port Is a designated communication endpoint in a computer, identified by a numerical code. It acts as a virtual gateway, enabling different applications or services to send and receive data efficiently. Port numbers help organize and manage the flow of information across computer networks. Each computer have 65535 possible network ports, however some of them are already reserved like the port 22 for SSH connections.

Over-Fetching Refers to a situation in which a software system retrieves more data from a server or database than is actually necessary for a particular task, operation, or view. This can lead to inefficiencies in terms of bandwidth usage, increased latency, and a potential waste of resources.

SDK A SDK is a set of tools for third-party developers to use in producing applications using a particular framework, platform or technology, normally provided by the owner company or development team.

Under-Fetching Happens when a system fails to retrieve enough data in a single request, leading to the need for additional requests to fulfill the requirements of a particular feature or view.

Vection Illusion A perceptual phenomenon in which a person perceives self-motion or movement while being stationary. Commonly experienced in virtual reality or other immersive environments, vection illusion occurs when visual cues, such as scrolling or moving visuals, lead the brain to interpret a sense of motion, even when the body is not physically moving. This illusion can sometimes result in sensations of dizziness or nausea.

Web API A set of rules and protocols that allow different software applications to communicate with each other over the internet. Web APIs enable the exchange of data and functionality between systems.

Websocket (WS) Room/Channel A communication space within a WebSocket connection where multiple clients can exchange real-time messages and data. It enables to group clients and privatize messages. It enables efficient and low-latency data sharing in web applications.