1 2 9 0

UNIVERSIDADE Đ
COIMBRA

Ricardo Luís Neves dos Santos

AUGMENTED REALITY PLATFORM - PHASE 2

Dissertation in the context of the Master in Informatics Engineering, specialization in Software Engineering, advised by Professor Marco Simões and Álvaro Menezes and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September of 2022

1 2 9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Ricardo Luís Neves dos Santos

Augmented Reality platform - Phase 2

Dissertation in the context of the Master in Informatics Engineering, specialization in Software Engineering, advised by Prof. Marco Simões and Álvaro Menezes and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September 2022

1 2 9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ð
COIMBRA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Ricardo Luís Neves dos Santos

Augmented Reality platform - Phase 2

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Engenharia de Software, orientada pelo Professor Doutor Marco Simões e Álvaro Menezes e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro 2022

Acknowledgements

I would like to start by expressing my gratitude towards my family for all the support they have provided throughout my long University years. Without them, I wouldn't be able to overcome so many obstacles and hardships.

For all my friends, a huge thank you, because whenever I would feel down you gave me the strength to rise and hold myself again. I'm forever grateful for those who helped through these years.

I want to thank the Critical Team, specially Álvaro, Rui, Miguel, and Carlos, for the support and advice given throughout the project and for being so welcoming. Any of this wouldn't certainly be possible without your help.

Finally, to my advisor, Professor Marco Simões, my sincerest gratitude for providing me the best advice throughout this internship and for always being available to help in any circumstances.

Abstract

Critical Software has several buildings with different meeting rooms around the world. Many of its employees need to travel between the different existing offices to attend meetings. In addition, the company frequently hires new employees that never walked into its offices. Determining the location of each building and its associated rooms proves to be a challenge that can be solved by using Augmented Reality.

The purpose of this internship is to provide a solution to Critical's employees, allowing them to improve their experience within the company's buildings using AR. To accomplish such objective, this internship focused on developing a system including a mobile application and the necessary back-end functionalities. This system includes indoors navigation assistance and allows its users to be guided through virtual signage. It also provides AR experiences resembling exhibitions of 3D virtual objects.

This report covers the many stages of the software development process, beginning with an assessment of the state of the art and progressing with the system description, the project planning and the methodology in use. It also includes all the choices and steps taken during development and testing.

Keywords

Augmented Reality, Location-based AR, AR Foundation, Mobile App

Resumo

A Critical Software possui vários edifícios com diferentes salas de reuniões por todo o mundo. Muitos de seus funcionários precisam de se deslocar entre os diferentes escritórios existentes para participar em reuniões. Para além disso, a empresa contrata novos funcionários com alguma frequência que precisam de se adaptar ao espaços de trabalho. A localização de cada edifício e das suas salas revela por isso ser um desafio que pode ser resolvido usando Realidade Aumentada.

O objetivo deste estágio é fornecer uma solução aos colaboradores da Critical, permitindo-lhes melhorar a sua experiência nos edifícios da empresa com recurso à Realidade Aumentada. Para atingir tal objetivo, este estágio focou-se no desenvolvimento de um sistema que inclui uma aplicação móvel e as funcionalidades de back-end necessárias. Este sistema possui funcionalidades como assistência à navegação indoors e permite que os seus utilizadores sejam guiados dentro de edifícios recorrendo a sinalização virtual. Também fornece experiências de Realidade Aumentada semelhantes a exposições de objetos virtuais 3D.

Este relatório abrange as várias etapas do projeto, começando com uma avaliação do estado da arte e progredindo com a descrição do sistema, o planeamento e a metodologia em uso. Para além disso também inclui todas as escolhas, etapas e testes realizados durante o desenvolvimento.

Palavras-Chave

Augmented Reality, Location-based AR, AR Foundation, Mobile App

Contents

Acronyms

3D Three Dimensional.

API Application Programming Interface.

AR Augmented Reality.

DTO Data Transfer Objects.

GPS Global Positioning System.

HMD Head Mounted Display.

MR Mixed Reality.

QR Quick Response.

SDK Software Development Kit.

TRL Technology Readiness Level.

VR Virtual Reality.

List of Figures

List of Tables

Chapter 1

Introduction

Critical Software has several buildings with different meeting rooms around the world. Many of its employees need to travel between the different existing offices to attend meetings. In addition, the company frequently hires new employees that never walked into its offices. Determining the location of each building and its associated rooms proves to be a challenge that can be solved by using Augmented Reality (AR) technology.

AR is a technology that allows us to add some degree of virtualization to the way we see the real world. It helps with solving problems and having possible applications in several areas such as Education, Health, Marketing and many others. It also provides different interactive experiences to its users by allowing interactions with several virtual elements. As this technology keeps growing with hardware advancements, so does its commercial value. For that reason, researching and bringing more knowledge on AR to the company might also have a greater impact in the near future.

Last year, Critical already had a project done using AR (ARCritical) during an internship. The resulting system is used to promote events within the company.

1.1 ARCritical Version 1.0

This section explains how the previously existing system works, its architecture and its functionalities.

1.1.1 Architecture

The previously developed system contains three main components: the Backend, a Web Interface, and a Mobile Application as we can see in figure 1.1.

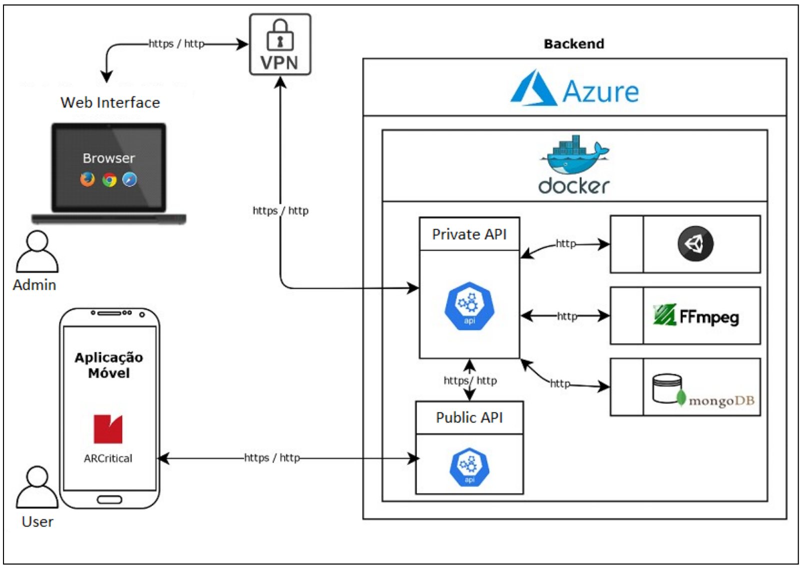


Figure 1.1: ARCritical's first version System Schematic. (Provided by Critical)

The **back-end** runs on several Docker containers, each of them having a specific function. The back-end of the system is then separated into a private Application Programming Interface (API), a public API and three other containers responsible for managing asset bundles and converting videos to unity supported formats and for storing all the necessary data. The private API is based on the Onion Architecture and was developed using the .NET Core 3.1 framework. It is used by the Web Interface and the public API. The public API provides access to functionalities available to users of the mobile application (downloading content from the application) and is used as a proxy (security measure).

The **Web Interface** uses the private API and is used to manage the content that appears in the mobile application.

The **Mobile Application** allows users to experience with AR content.

1.1.2 Functionalities

To further explain how the previous system works, it is necessary to clarify some concepts used in the application:

- **Campaigns:** A set of Trigger-Action pairs that remain available as application content for a given period of time.
- **Triggers:** Images that are essentially markers, which can be identified (having a rating so we know how easy it is to identify) and, consequently, trigger Actions.
- **Actions:** Content that is displayed once the application identifies a trigger. An Action can be an audio file, an image, with or without background, a video, with or without background, or even a Three Dimensional (3D) model.

Web Interface

The Web Interface is intended to be used by an authorized user to manage the mobile application content, specifically Campaigns, Triggers and Actions as seen in figure 1.2.

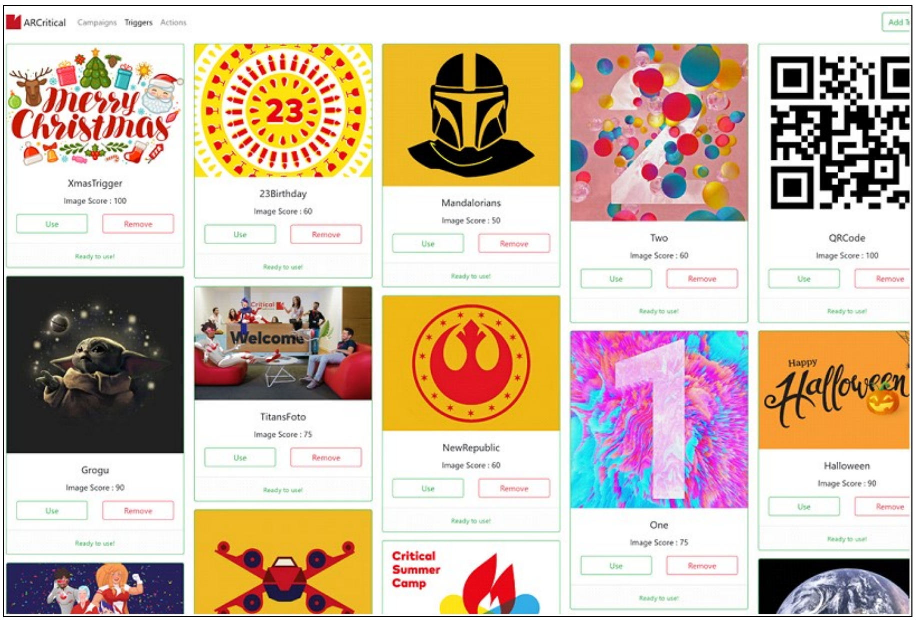
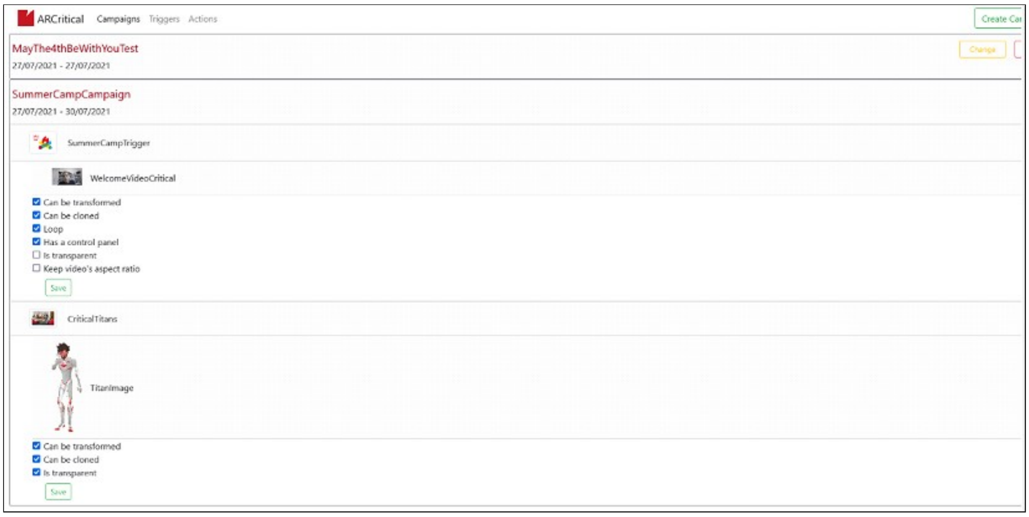


Figure 1.2: Web Interface. (Provided by Critical)

Through this interface, the user can add, remove, change any trigger, action and associate or disassociate both with any existing campaign demonstrated in figure 1.3. It is also possible to change any available property of each one of them (e.g. changing the end date of a campaign).



MayThe4thBeWithYouTest
27/07/2021 - 27/07/2021

SummerCampCampaign
27/07/2021 - 30/07/2021

SummerCampTrigger

WelcomeVideoCritical

☑ Can be transformed
☑ Can be cloned
☑ Loop
☑ Has a control panel
☐ Is transparent
☐ Keep video's aspect ratio

Save

CriticalTitans

TitanImage

☑ Can be transformed
☑ Can be cloned
☑ Is transparent

Save

Figure 1.3: Web Interface - Campaigns. (Provided by Critical)

Mobile Application

The mobile application can be used by any Critical employee. It's the part of the system that contains AR technology, allowing the user to experiment with it. Through the application, the user can point the camera to a trigger, receiving AR content available from the action associated with that specific trigger. An example of an action would be a 3D model as the user can interact with it by placing it in other locations around him as in figure 1.4.

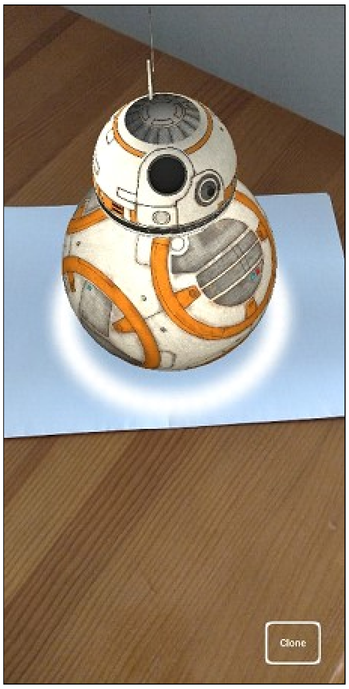


Figure 1.4: Mobile Application - 3D model. (Provided by Critical)

1.2 Objectives

The purpose of this internship is to provide a solution to Critical's employees, allowing them to improve their experience within the company's buildings using AR. To accomplish such objective, this internship focused on developing a system including a mobile application prototype and the necessary back-end functionalities. This system has to include indoors navigation assistance and should also allow its users to be guided through virtual signage. It also should provide AR experiences resembling exhibitions of 3D virtual objects. In addition to these characteristics, all content should be configurable and changeable without requiring a new build and installation of the mobile application.

1.3 Document's Structure

In the second chapter, a number of concepts related to AR are presented. In addition, a summary of the history of AR is also given. Following these sections there's a presentation of possible applications and future trends of the technology and a summary of the analysis made to similar AR applications. Finally, there is also a summary of the studied technologies considered for the development of this project.

The third chapter contextualizes the scope of the project starting by explaining some of the project related concepts and defining use cases. Following these sections a set of functional and non-functional requirements is shown as well as the project's constraints. At the end of the chapter, there's the section where the architecture and the implemented functionalities are explained.

The fourth chapter goes into detail on the development methodology, all the necessary tools and planning done for the project. At the end of it, there's also a section dedicated to risk management.

The fifth chapter explains details related to this project's development. It also explains some conducted tests and its results.

Finally, in the conclusion, there's a summary about the work done as well as the future possibilities.

Chapter 2

State of The Art

The first concept similar to that of the AR as we know today ever documented in science fiction appeared in the book "The Master Key" by L. Frank Braum in 1901 (figure 2.1). It consisted of a pair of spectacles that allowed the wearer to see a mark upon the forehead of people with a letter indicating his or her character. This character marker has been considered to be an early foreshadowing of features analogous to those obtainable in AR devices [57].

Figure 2.1: L. Frank Baum's 1901 novel, The Master Key. Reference: [57]

AR is a technology that allows seeing real-life environments enhanced with a digital augmentation and computer-generated overlay. This concept of AR has been evolving since there's been the necessity to bridge AR with Virtual Reality (VR), that being considered Mixed Reality (MR) [38].

This technology has become increasingly popular in modern-day system development and has potential applications in different areas such as entertainment and marketing. The current document's section will tackle the evolution of AR throughout history, its future trends and its necessary hardware and associated development tools and concepts. It will also define the current developed system in which modifications are necessary to accommodate the new functionalities.

2.1 Augmented Reality Concepts

MR is the merging of real and virtual worlds to produce new environments. It's a spectrum as in figure 2.2 where physical and digital objects co-exist and interact in real-time. AR and VR are included in this spectrum [73].

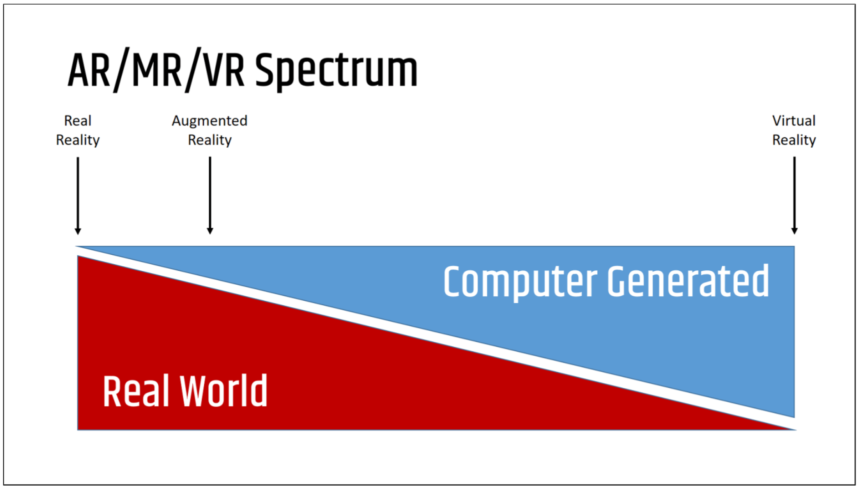


Figure 2.2: Mixed Reality Spectrum. Reference: [64]

AR is a technology that allows the perception of a different version of the real world. This is achieved through the use of digital visual elements, sounds, or other sensory stimuli. There are two different types of AR: Marker-based AR and Markerless AR.

Marker-based AR [3; 12] allows the delivery of information linked to a specific object rendering digital visualizations. While examining an image, Quick Response (QR) Code, or pattern presented upfront, it provides information about it on the screen.

The marker is usually a distinctive picture that is recognized. Images with corners and edges or high contrast are usually the best suited for it. Marker-based AR offers real-time user interactions, having minimum production costs and working on low-end devices.

Markerless AR [3; 12] works by using sensors (as digital compasses or positioning sensors) for reading data of any physical place while predicting the area where the user is navigating. An example of markerless AR would be an application that can place virtual furniture inside the user's living room, allowing different combinations of objects, styles, and locations. Markerless AR offers more stability in user experience as it allows moving digital objects freely. It also provides more capabilities than Marker-based AR, however, usually requiring more hardware.

Location-based AR [3; 12] is a type of a Markerless AR system. Location-based AR doesn't require any target or markers to execute the rendering of the AR elements or to identify where to put virtual objects. It usually relies on location-based AR technology using Global Positioning System (GPS) data and digital compasses to define the user′s device location and position.

There are other important concepts related to AR that should be clarified:

Occlusion - Occlusion [7; 10; 107] is when an object in a 3D space blocks the view of another object. Occlusion is important to ensure a high level of realism and allow the objects to behave as they would under normal conditions.

Anchor - Anchors [7; 10] make objects appear to stay in the same position and orientation in space assisting with the illusion of virtual objects in the real world.

Cloud Anchor - Cloud Anchors [7; 10] behave similarly to anchors. Because they are hosted in a cloud service, users can view and interact simultaneously with these objects from different positions.

Oriented points – An oriented point [7; 10] is a type of point that contains coordinates representing the positions and a virtual endpoint to specify an orientation vector. It allows placing virtual objects on angled surfaces.

Point Cloud – Point cloud [7; 10; 77]is a set of data points in space that may represent a 3D shape or object.

Positional Tracking - Positional tracking [7; 10; 107] allows a device to determine its position relative to its environment.

SLAM - SLAM stands for Simultaneous Localization and Mapping [7; 10]. This is a technology used to interpret the real environment to place 3D objects within it.

2.2 History of Augmented Reality

In 1961, Philco Corporation created the first head-mounted display, called the "Headsight" (figure 2.3), which had a screen and a tracking system linked to a closed circuit TV [101].

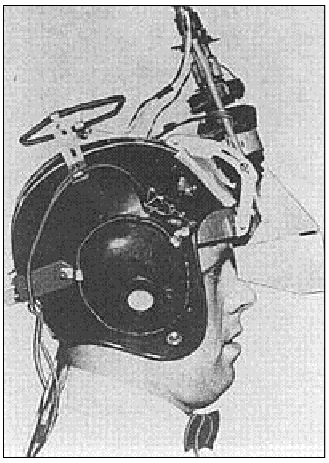


Figure 2.3: Philco Headsight. Reference: [101]

In 1962, Hughes Aircraft Company revealed the “Electrocular”, an head-mounted monocular display that reflected a TV signal in to a transparent eyepiece shown in figure 2.4. These were made for military use and display of information[35; 68; 81].

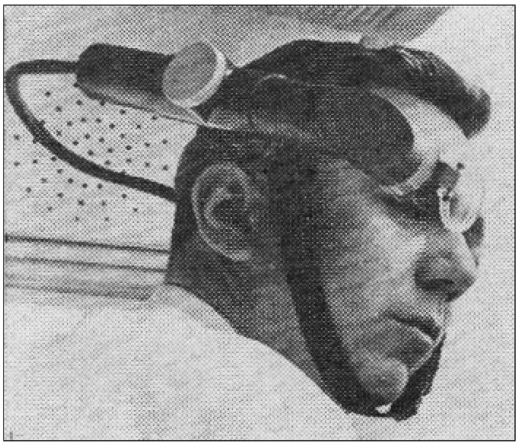


Figure 2.4: The "Electrocular". Reference: [68]

In 1968 Ivan Sutherland with the help of his students Bob Sproull, Quintin Foster, and Danny Cohen built a Head Mounted Display (HMD) system with monochrome monitors and mirrors [47; 76]. The half-silvered mirrors enabled the observer to see through the computer graphics display into the real world blending high-resolution 3D graphics with a view of the real world. It’s mechanical tracking system was named “The Sword of Damocles” and is considered today to be the first AR HMD system (figure 2.5).

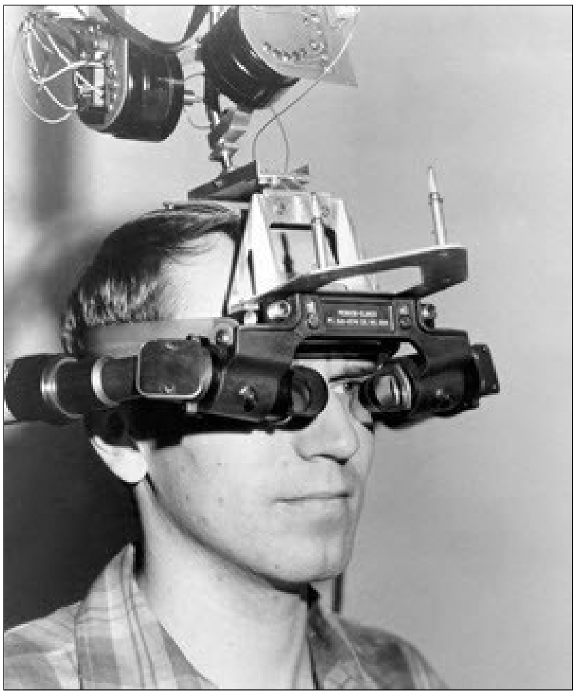


Figure 2.5: Dr. Ivan Sutherland’s head-mounted 3D display. Reference: [47]

In 1975, Myron Krueger developed “Videoplace” which allowed its users to interact with virtual objects in real-time through different gestures [21; 34; 65] as explained in the schematic of figure 2.6.

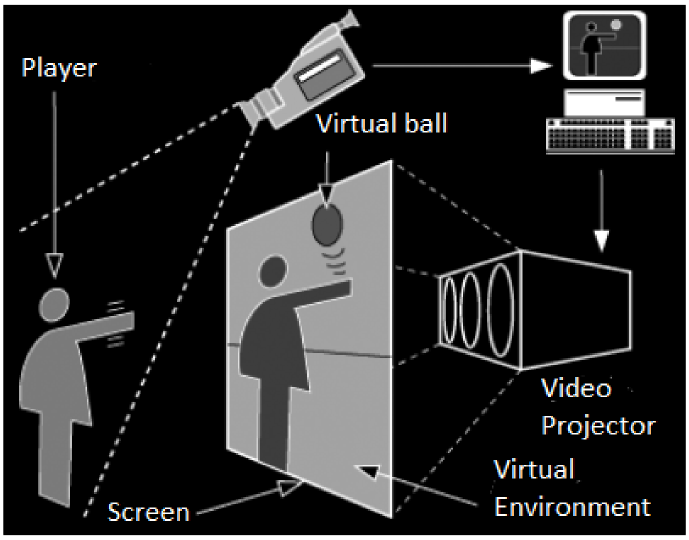


Figure 2.6: Videoplace’s Illustration. Reference: [34]

Later in 1980, Steve Mann, a computational photography researcher, created the "WearComp" that included an antenna to communicate wirelessly and share video [100]. He continued to develop it as shown in figure 2.7.

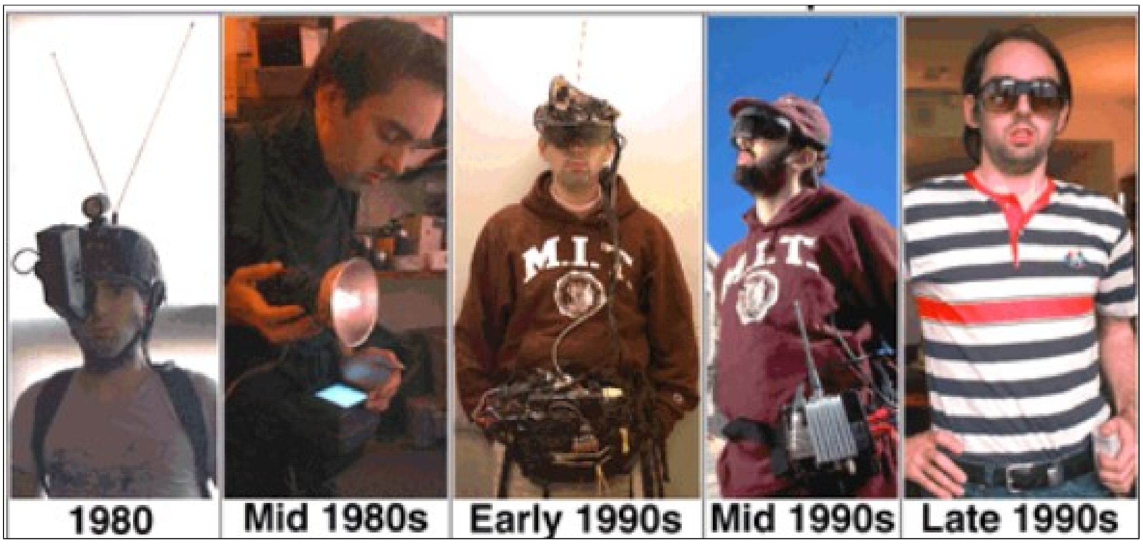


Figure 2.7: Evolution of Steve Mann’s "wearable computer". Reference: [100]

The term of “augmented reality” finally appeared in the work of Caudell and Mizell at Boeing in 1992. Their work displayed wire bundle assembly schematics in a HMD in order to assist airplane factory workers (figure 2.8) [47; 55].

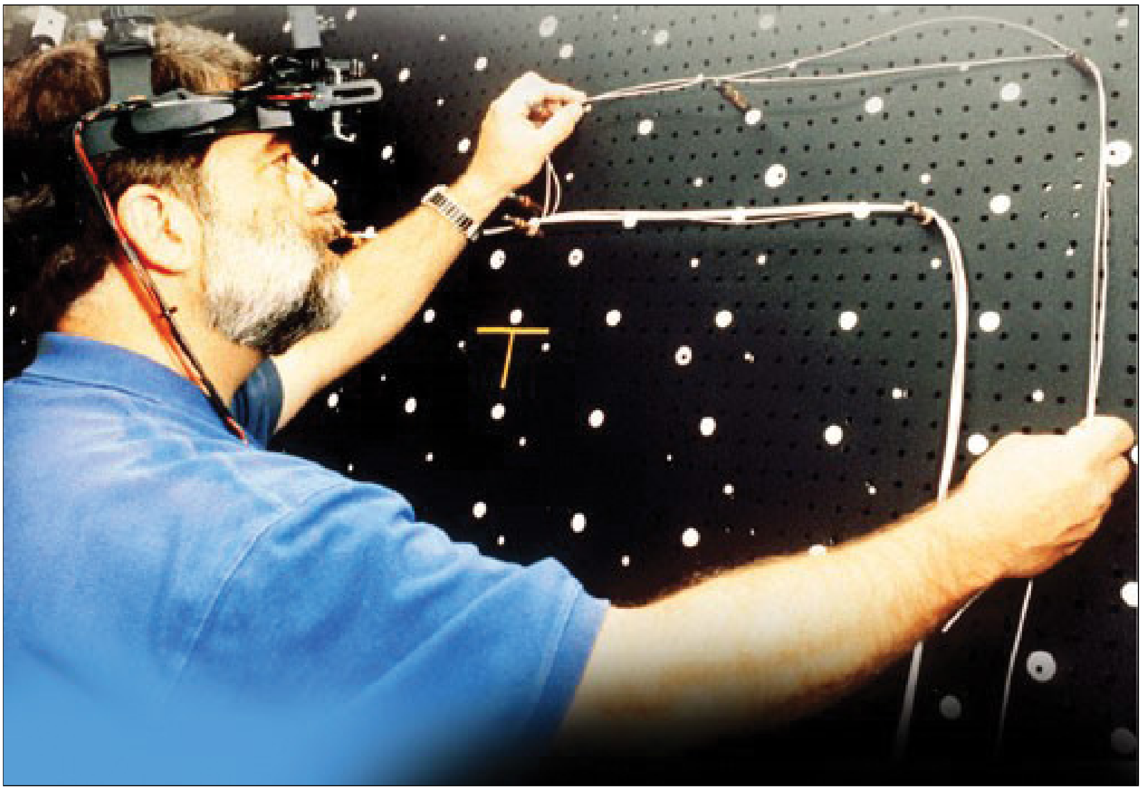


Figure 2.8: Researchers at Boeing using a see-through HMD to guide the assembly of wire bundles. Reference: [47]

In 1992, Louis Rosenburg, created 'Virtual Fixtures', which was one of the first fully functional AR systems. The system allowed military personnel to virtually control and guide machinery to perform tasks like training their United States Air Force pilots on safer flying practices [56; 65; 70] as we can see in figure 2.9.

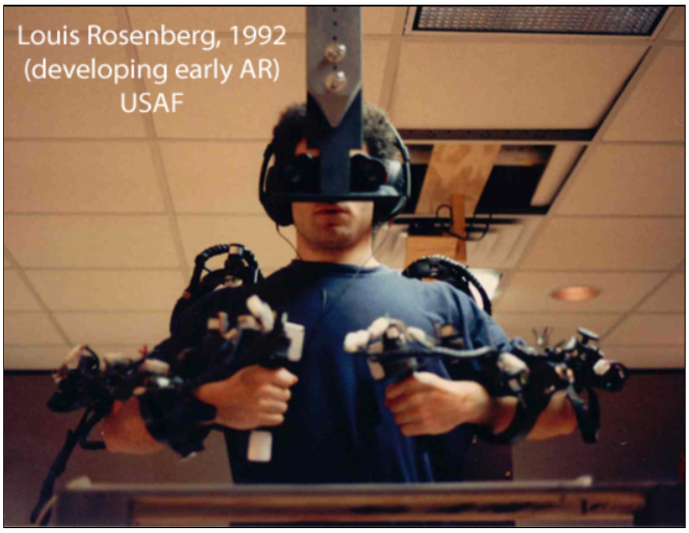


Figure 2.9: Louis Rosenberg testing Virtual Fixtures. Reference: [56]

In 1997, the first mobile AR system, "Touring Machine" (figure 2.10) was developed in the Computer Graphics and User Interfaces Lab of Columbia University by Steven Feiner. This system used an HMD, an handheld display and a computer with GPS and Internet connection [75].

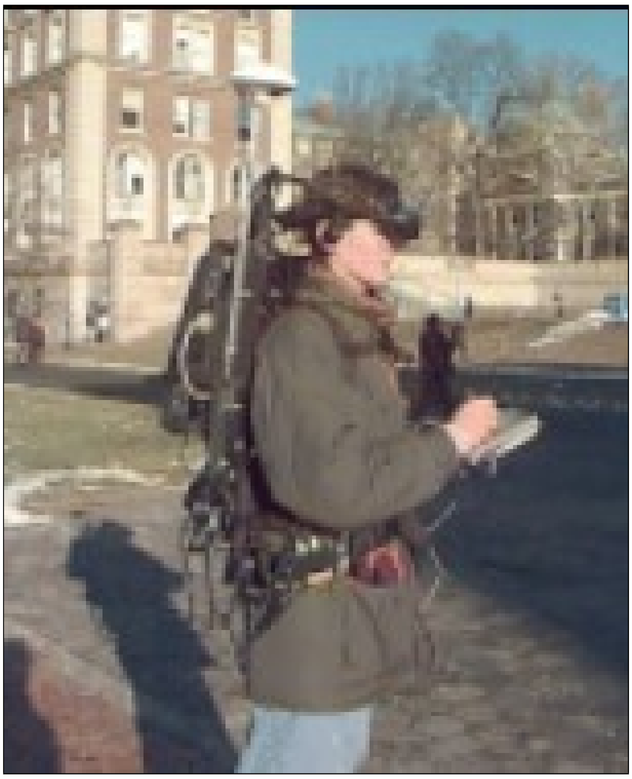


Figure 2.10: MARS Touring Machine's Prototype being tested. Reference: [75]

In 1998 the virtual 1st & Ten graphic system was broadcasted the first live National Football League game. The technology displays a yellow line overlaid on top of the feed as figure 2.11 shows [32; 49].

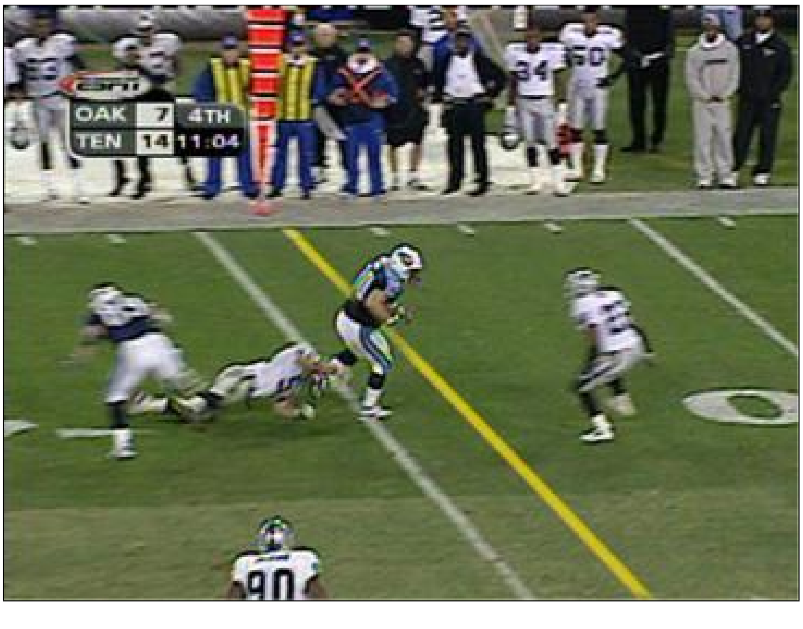


Figure 2.11: 1st & Ten graphic system being used for the first time. Reference: [102]

In 1999, the first AR software kit emerged, the AR Toolkit. Its first open-source version was released through the Human Interface Technology Laboratory in 2001 [11].

The first AR game named ARQuake appeared in the year of 2000 (figure 2.12). It uses a Toshiba Pentium-233 laptop running Linux, driving an I-Glasses color PAL display, along with a Precision Navigation TCM2-80 orientation sensor, and Garmin GPS12XL with GPS for positioning [60; 82].

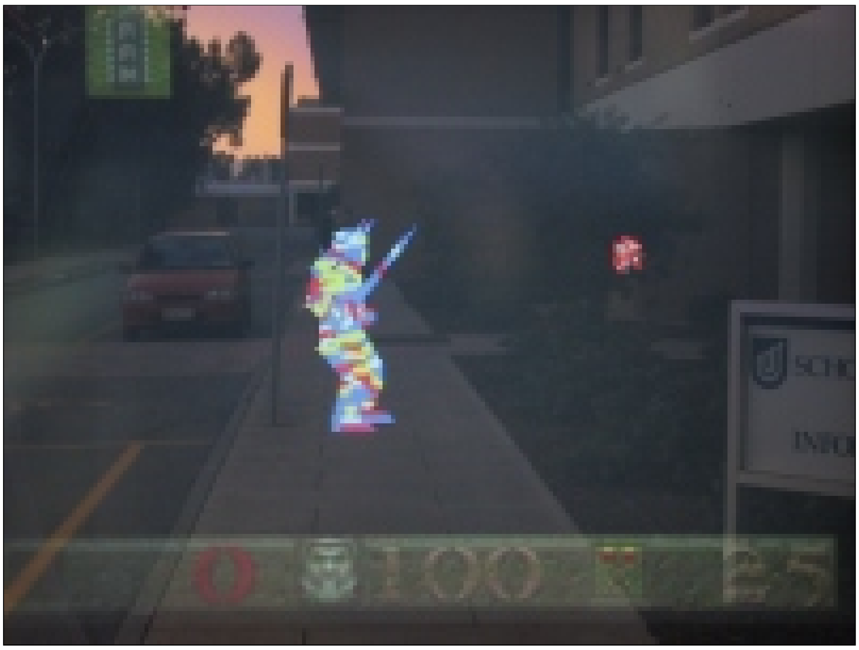


Figure 2.12: ARQuake first person view. Reference: [82]

In 2005 Anders Henrysson using the ARToolKit developed AR-Tennis (shown in figure 2.13), a multiplayer game for Nokia mobile users [43; 46].

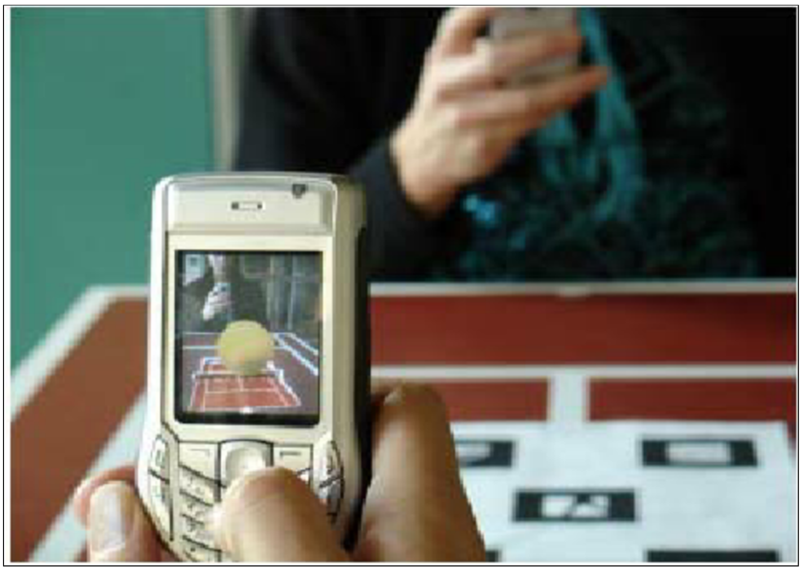


Figure 2.13: AR-Tennis being played. Reference: [43]

One year later, Gerhard Reitmayr and Tom W. Drummond made a new system for Outdoors AR using sensors such as a gyroscope, inertial sensors and magnetic sensors [67].

In 2013, Google launched its brand of smart glasses, Glass, consisting of an optical head-mounted display made available to the public in 2014 [42; 71]. Its first model is displayed in figure 2.14.

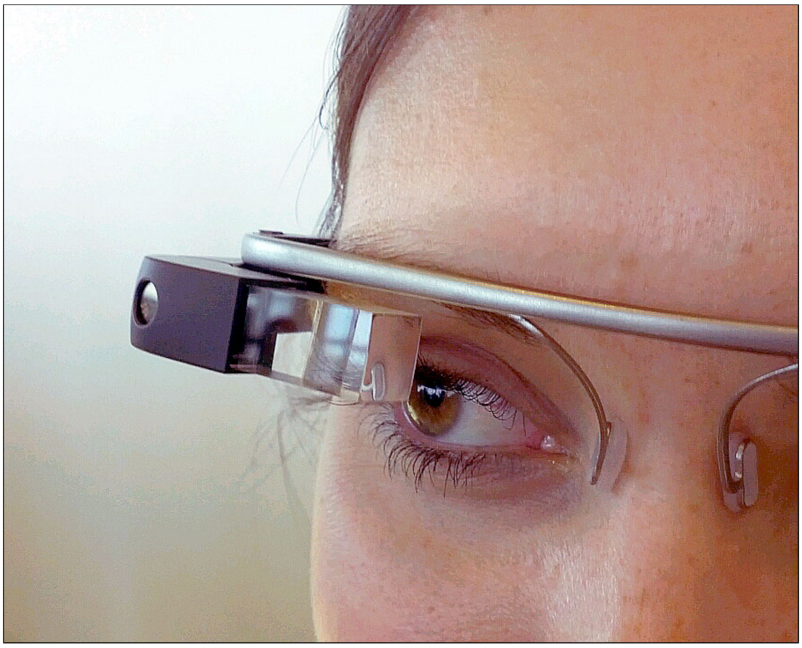


Figure 2.14: Google Glass. Reference: [103]

Three years later, Microsoft released a pair of mixed reality smartglasses, the Hololens (figure 2.15)[52].

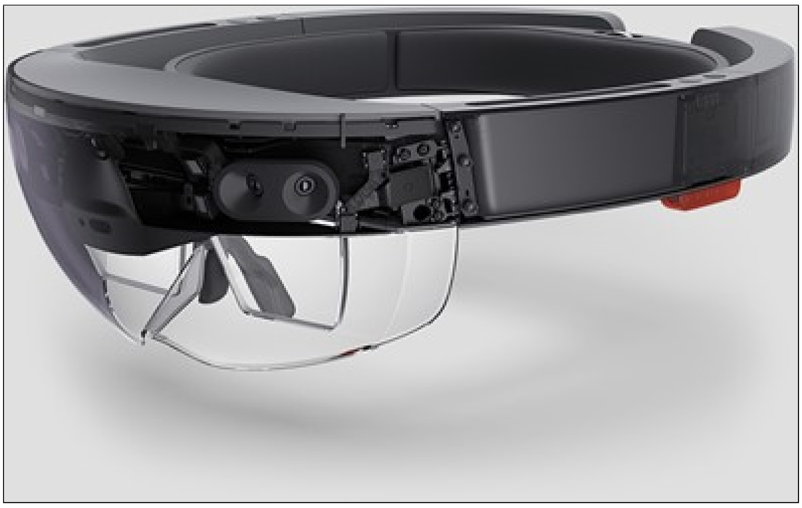


Figure 2.15: Hololens. Reference: [52]

In 2017, Apple launched the ARKit Software Development Kit (SDK) for iOS systems making the AR mobile development possible for iOS [24]. Google also launched the ARCore SDK for Android systems. ARCore facilitated the mobile development of AR applications for Android [61].

One year later, Google also announced the WebXR Device API for both AR and VR [8].

In Portugal, the "Associação Portuguesa para o Desenvolvimento das Comunicações" [4] has currently an AR/VR section that promotes and encourages the development and the collaboration between companies on this technology.

2.3 Areas and Future Trends

2.3.1 Entertainment and Gaming

In June 2021, the "Muséum national d'Histoire naturelle" [27; 28] in Paris launched an AR experience using Microsoft's Hololens. The project, called "Revivre," let visitors encounter digital animals that are already extinct in the real world as shown in figure 2.16.

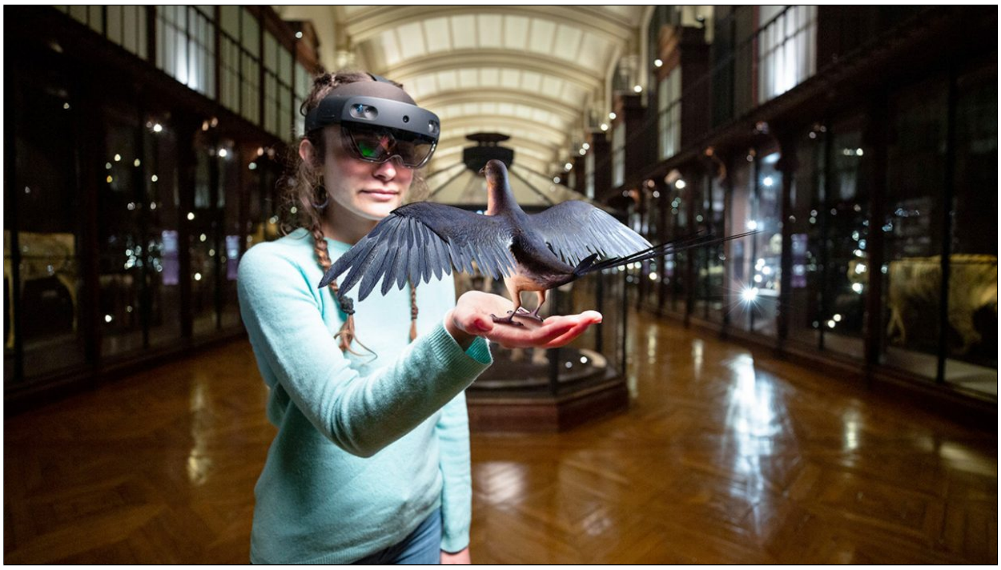


Figure 2.16: At "Revivre", the French National Museum of Natural History. Reference: [28]

In the Gaming industry, the use of AR technology has been growing in the last years. In 2016, Niantic and Nintendo launched Pokemon Go (figure 2.17), being the most popular mobile AR game until today. Other games were launched as well, such as Ingress Prime and Jurassic World Alive [74].

Figure 2.17: Pokemon Go. Reference: [74]

Even on television programs this technology has been used, either in sports events or in event's transmissions. AR has also been present in several online streaming events such as the ceremony of several League of Legends competitions as we can see in figure 2.18[80].

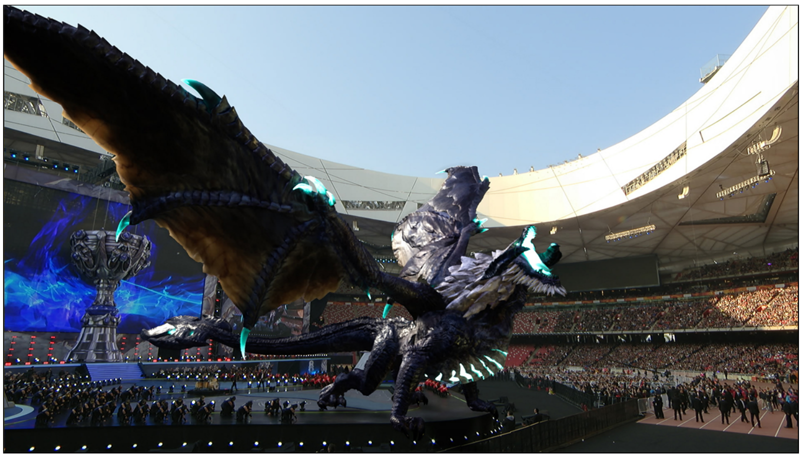


Figure 2.18: League of Legends competition opening ceremony. Reference: [22]

2.3.2 E-commerce and Marketing

In the last couple of years, although being considered an emerging technology, several companies have been adopting AR to provide a better shopping experience to their clients. The bet on AR has been driving sales for companies like IKEA (figure 2.20), The Cambridge Satchel Company (figure 2.19), Amazon, Alibaba, Zara, Lacoste and many others [72].

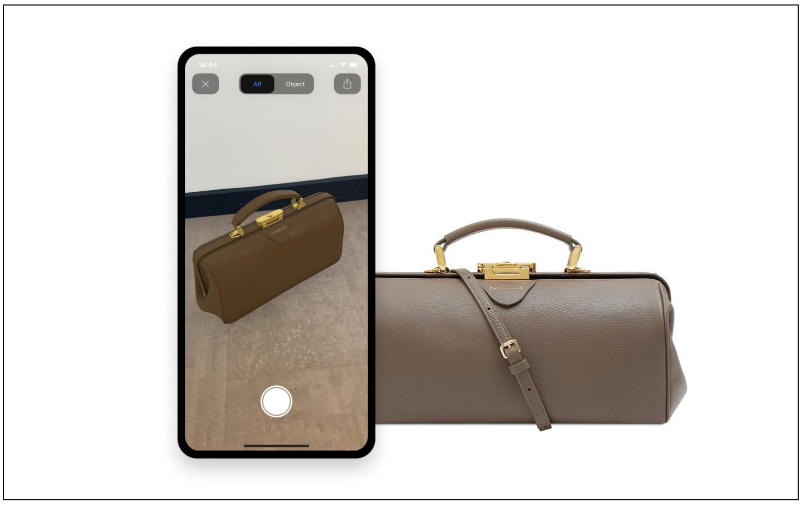


Figure 2.19: Cambridge Satchel Company app. Reference: [31]

According to Google's 2019 AR Survey [41], 66% of people are interested in using AR for help when making purchasing decisions, and this number tends to grow due to the equipment update and the mature technology allowing for a better user experience.

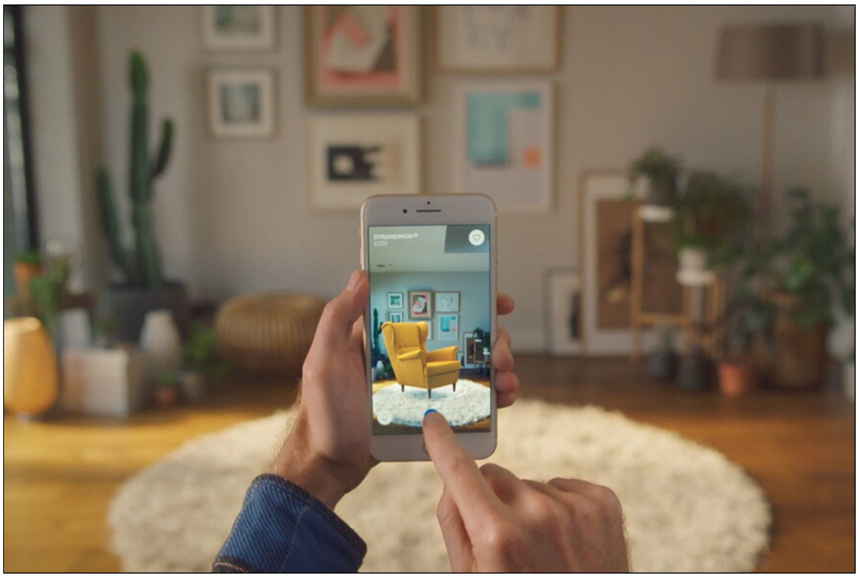
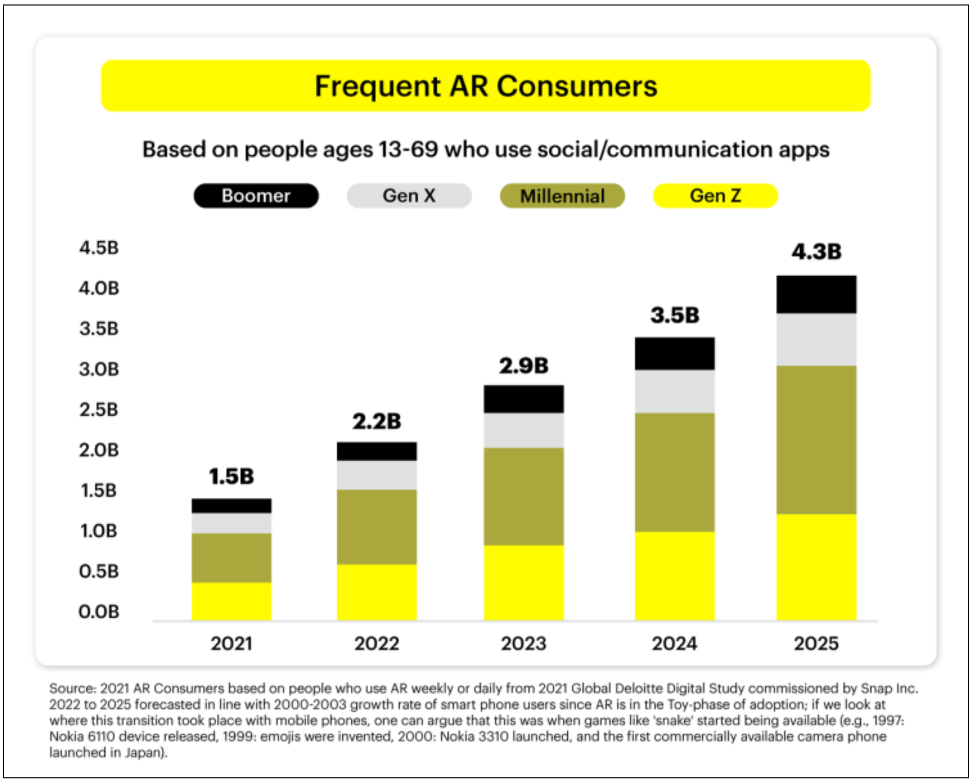


Figure 2.20: IKEA app. Reference: [45]

Snap [36] also commissioned a study from Deloitte Digital and talked to 15,000 consumers across several countries. According to their results, by 2025, nearly 75% of the global population and almost all smartphone users will be frequent AR users. The estimated number of AR consumers by generation is shown in figure 2.21.



Source: 2021 AR Consumers based on people who use AR weekly or daily from 2021 Global Deloitte Digital Study commissioned by Snap Inc. 2022 to 2025 forecasted in line with 2000-2003 growth rate of smart phone users since AR is in the Toy-phase of adoption; if we look at where this transition took place with mobile phones, one can argue that this was when games like 'snake' started being available (e.g., 1997: Nokia 6110 device released, 1999: emojis were invented, 2000: Nokia 3310 launched, and the first commercially available camera phone launched in Japan).

Figure 2.21: AR Users. Reference: [36]

2.3.3 Healthcare and Education

AR [26; 94] has been an active area of research in an educational setting for learning and teaching activities (figure 2.22). It has been attracting a lot of interest in the research community because it provides unique learning experiences that cannot be achieved using other technologies. Furthermore, AR can be more appealing than traditional learning methods, resulting in better engagement with the educational content.

Figure 2.22: AR being used for educational purposes. Reference: [94]

In Healthcare, [39; 44] AR has been an upcoming technology trend as well. It has been a tool that demonstrates to be effective in training medical students, displaying vital information about patients, or even in surgeries as shown in figure 2.23.

Figure 2.23: Surgery with AR. Reference: [44]

2.4 AR Applications Analysis

The main goal of this section is to help understanding key features and technologies used in popular AR applications. It will also give information regarding some of the characteristics of other applications that may be incorporated into the project.

Magical Park is one of many AR mobile games that use outdoors AR. Similarly to Pokemon Go, this game uses the device's GPS location and its camera capabilities to place digital objects resembling various creatures.

Google Maps has a Live View function that uses AR to help its users navigate while walking. Once a destination is selected, the app gives directions in voice instructions, while prompting the user to lift the phone and use the AR functions as shown in figure 2.24.

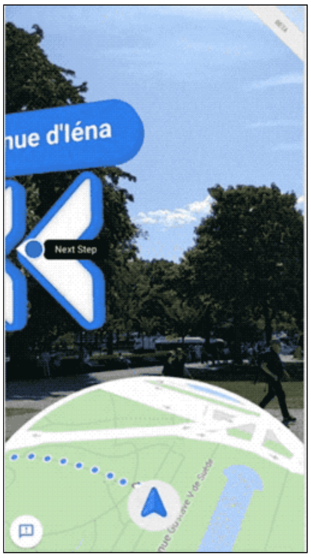


Figure 2.24: Google Maps Live View.

The Star Walk app allows users to point their devices toward the sky and be given the names of stars, constellations, and any visible planets. If the device is pointed at the ground, the names of stars and constellations of the opposite hemisphere are visible. This application also allows the user to give his coordinates manually or to use the device's GPS functions.

GuideBot (figure 2.25) is a mobile application that relies on AR markers and SLAM technologies to provide assistance while navigating inside a building.

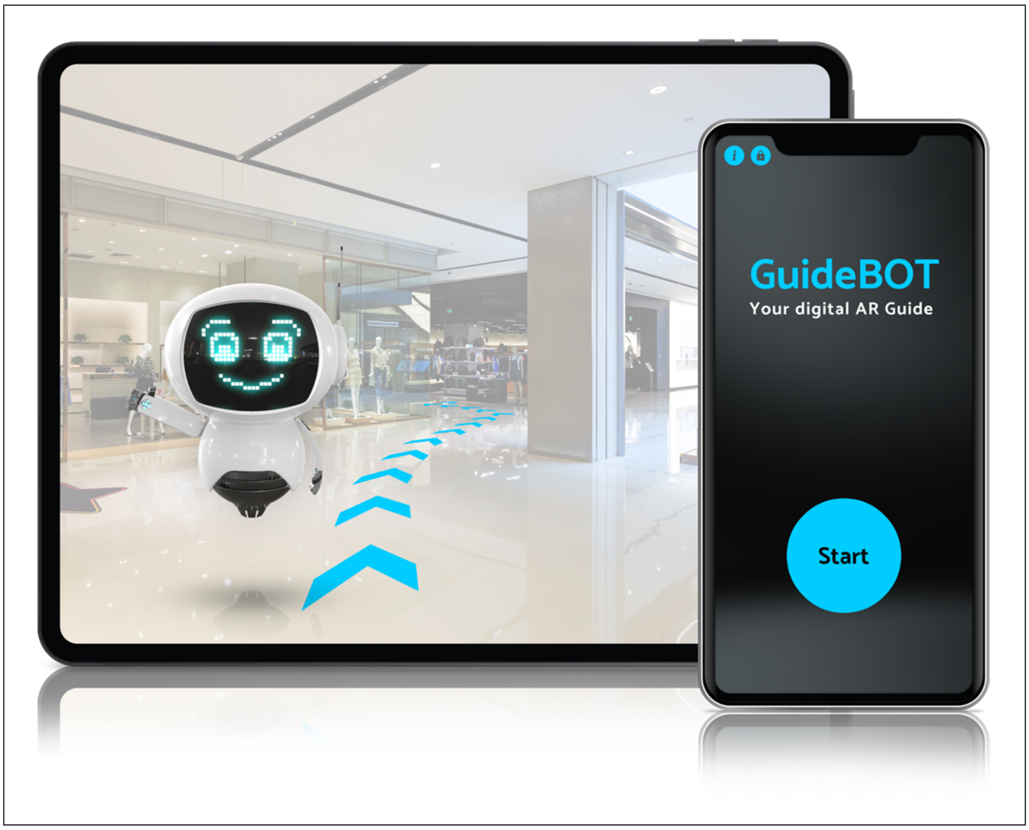


Figure 2.25: GuideBot application.

Some of the characteristics of these applications were considered to be of most importance for this project. The use of SLAM and markers is certainly one of them, since it's crucial to track a device's position as the GPS signal is not accurate enough and doesn't work well enough inside buildings. Occlusion is also important to display AR content in a realistic way, however it might not be feasible to include in this project since it relies on specific hardware. Other features of these applications, such as the mini-map in Google Maps, were also considered to be important.

2.5 Development Tools and Limitations

An AR SDK provides the necessary tools and functionalities to develop applications that apply this technology. The following 6 SDK's were the chosen after analyzing the most suitable solutions in the context of this project:

- **Vuforia**
- **ARCore**
- **ARKit**
- **AR Foundation**
- **Pikkart AR**
- **Wikitude**

Other support technologies, such as plugins were taken into consideration. To evaluate these technologies, a set of metrics and a list of necessary features were defined. The following are the criteria considered for further analysis:

- Price - Spending as few resources as possible;
- Community Size - Existing Forums and number of commits and repositories available for that technology;
- Functionalities - As it is of most importance to have Markerless AR and Marker-based AR functionalities available;
- Supported platforms - Android and iOS supported versions;
- Integration with the already existing system - Developed in Unity3D using AR Foundation;
- Available Documentation - Quality and quantity of documentation (qualification from 1 to 5);

2.5.1 ARCore

ARCore [6] is a free SDK that uses three key capabilities to integrate virtual content with the real world as seen through the camera:

- **Motion tracking** allows the phone to understand and track its position relative to the world.
- **Environmental understanding** allows the phone to detect the size and location of all type of surfaces: horizontal, vertical and angled surfaces like the ground, a coffee table or walls as used in figure 2.26.
- **Light estimation** allows the phone to estimate the environment's current lighting conditions.

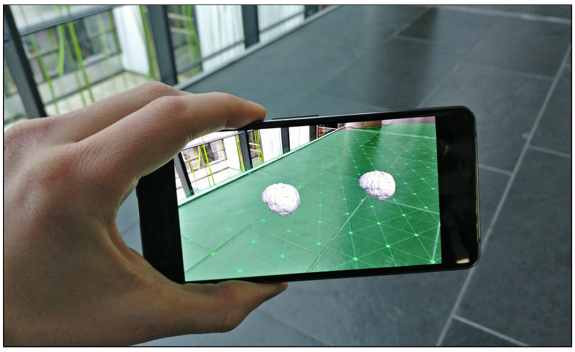


Figure 2.26: Plane visualization and object placement. Reference: [2]

ARCore also provides more functionalities such as :

- **Recording** - Start, stop, and check the status of an ARCore session recording.

- **Instant Placement** - The Instant Placement API allows the user to place AR objects instantly, without having to first move their device to allow ARCore to establish full tracking and detect surface geometry. After the user has placed an object, its pose is refined in real time as the user moves through the environment. Once ARCore is able to determine the correct pose in the region where the AR object has been placed, it updates the object's pose and tracking method.

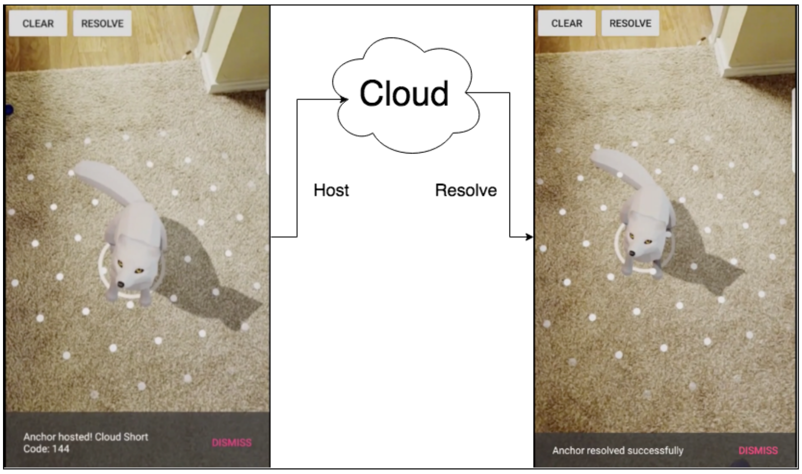
- **Depth API and Occlusion** - The Depth API can power object occlusion, improved immersion, and novel interactions that enhance the realism of AR experiences.

- **Lighting Estimation API** - analyzes given images for such cues, providing detailed information about the lighting in a scene. This information can be used when rendering virtual objects to light them under the same conditions as the scene they're placed in.

- **Augmented Faces API** - allows to render assets on top of human faces without using specialized hardware. It provides feature points that enable any app to automatically identify different regions of a detected face. These regions can then be used to overlay assets in a way that properly matches the contours of an individual face.

- **Anchor** - anchors are used to place objects in the scene.

- **Cloud Anchor** - are similar in behavior to anchors, and the basic guidelines for anchors also apply to Cloud Anchors (figure 2.27). Cloud Anchors differ from anchors in that they're hosted on the ARCore Cloud Anchor service. This hosting enables users to share experiences in the same app, allowing the creation of persistent anchors based on their location.



Anchor hosted! Cloud Short Code: 144

Anchor resolved successfully

Figure 2.27: Placement of Cloud Anchor. Reference: [51]

Currently ARCore does not support the tracking of 3D objects. ARCore is designed to work on a wide variety of qualified Android phones running Android 7.0 (Nougat) and later. Cloud Anchors and Augmented Faces are supported on all ARKit-compatible devices running iOS 11.0 or later.This SDK has well-written and organized documentation and a repository with several sample apps in different programming languages. From all of the considered SDK's, ARCore was the one with the largest community.

2.5.2 ARKit

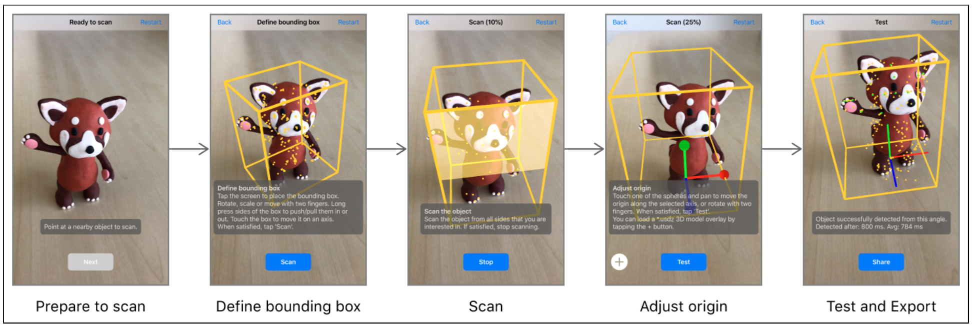
ARKit [9] is an Apple SDK (also free) that enables the development of AR apps. It provides similar functionality to ARCore, however it includes 3D object detection and more. It provides the following functionalities:

- **Depth API** - The advanced scene understanding capabilities built into the LiDAR scanner enable this API to use per-pixel depth information about the environment. LiDAR is technique used to create high-resolution three-dimensional models of landscapes. LiDAR uses laser pulses to measure distances to the ground and other features. Combined with the 3D mesh data generated by Scene Geometry, this depth information makes the occlusion of virtual objects more realistic by enabling instant placement of virtual objects and seamlessly blending them with their physical surroundings. This enables new features such as taking precise measurements and applying effects to the user's environment.

- **Instant AR** - The LiDAR scanner enables rapid detection of planes for immediate placement of AR objects in the real world without scanning. Instant placement of AR is automatically enabled on iPhone 12 Pro, iPhone 12 Pro Max and iPad Pro for all apps created with ARKit.

- **Motion Capture** - Capture a person's movements in real time with a single camera by understanding body position and movement as a series of joints and bones. Height estimation improves on iPhone 12, iPhone 12 Pro, and iPad Pro in all apps built with ARKit.

- **Scene Geometry** - Creates a topological map of the space with labels identifying floors, walls, ceilings, windows and more.
- **People Occlusion** - The content of AR moves realistically behind and in front of people in the real world, making AR an immersive experience while enabling green screen effects in almost any environment.
- **Object Detection** - ARKit creates reference points of objects by scanning them. These points can later be used to identify these objects as demonstrated in figure 2.28.



Point at a nearby object to scan.

Define bounding box
Tap the screen to place the bounding box. Rotate, scale or move with two fingers. Long press sides of the box to push/pull them in or out. Touch the box to move it on an axis. When satisfied, tap 'Scan'.

Scan the object
Scan the object from all sides that you are interested in. If satisfied, stop scanning.

Adjust origin
Touch one of the spheres and pan to move the origin along the selected axis, or rotate with two fingers. When satisfied, tap 'Test'.
You can load a *.usdz 3D model overlay by tapping the + button.

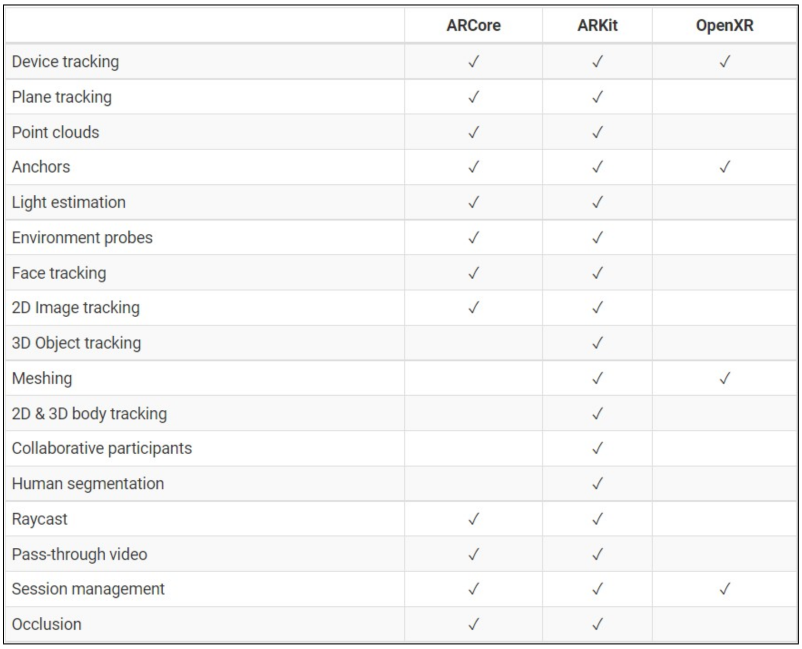
Object successfully detected from this angle.
Detected after: 800 ms. Avg: 784 ms

Figure 2.28: Object Detection using the ARKit. Reference: [5]

The Depth API is specific to devices equipped with the LiDAR Scanner (iPad Pro 11-inch (2nd generation), iPad Pro 12.9-inch (4th generation), iPhone 12 Pro, iPhone 12 Pro Max). All these functionalities are supported on iOS 11.0+ but some work has evolved on more recent versions. Similarly to ARCore, its documentation is clear and organized and there are several videos and forums.

2.5.3 AR Foundation

AR Foundation is a free cross-platform framework that allows AR applications development for Android or iOS devices in Unity3D. AR Foundation is built on subsystems. A "subsystem" is a platform-independent interface for representing various types of information. A provider is a concrete implementation of a subsystem. For example, the ARCore XR plugin package contains the ARCore implementation for many of the AR subsystems, or the ARKit XR plugin package, which contains ARKit. For that reason, this framework provides the same functionalities as ARCore and ARKit since it has both of them (figure 2.29). Some functionalities of ARCore or ARKit might not be available, but they can be incorporated using its Unity3D plugins. AR Foundation also provides OpenXR functionalities. OpenXR is a free, open standard that gives XR—platforms and devices like the Hololens2 and Oculus—high-performance access to AR and VR.



	ARCore	ARKit	OpenXR
Device tracking	✓	✓	✓
Plane tracking	✓	✓	
Point clouds	✓	✓	
Anchors	✓	✓	✓
Light estimation	✓	✓	
Environment probes	✓	✓	
Face tracking	✓	✓	
2D Image tracking	✓	✓	
3D Object tracking		✓	
Meshing		✓	✓
2D & 3D body tracking		✓	
Collaborative participants		✓	
Human segmentation		✓	
Raycast	✓	✓	
Pass-through video	✓	✓	
Session management	✓	✓	✓
Occlusion	✓	✓	

Figure 2.29: Functionalities offered by AR Foundation. Reference: [84]

2.5.4 Pikkart AR

Pikkart SDK [62] is another alternative available in the market that has compatibility with Unity3D through its own plugin.

This SDK provides the following functionalities:

- **Cloud recognition with API** - Provides cloud services to create and save AR elements in a scalable way, similarly to Cloud Anchors.
- **Image recognition** - Provides similar image recognition functionalities to other SDK's.
- **ARLogo** - Creates multiple AR experiences on different prints of the same image (figure 2.30).

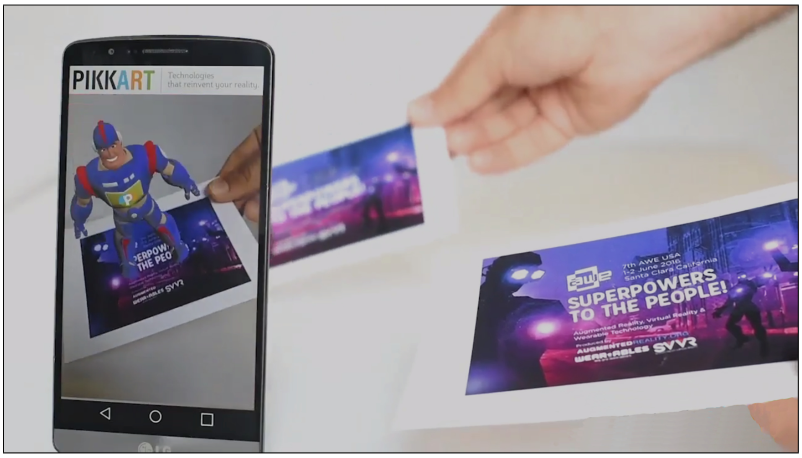
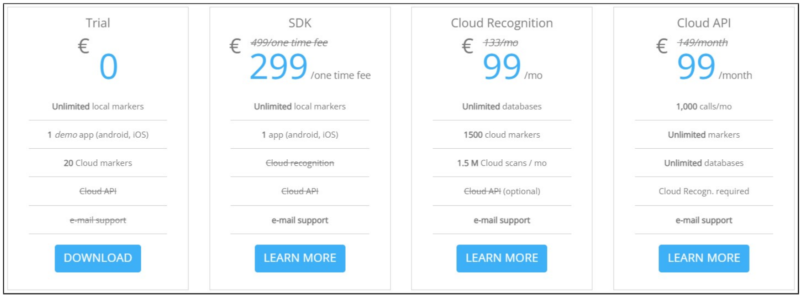


Figure 2.30: Pikkart ARLogo functionality in practice. Reference: [62]

- **Geo augmented marker** - Adds geolocated augmented markers, integrating navigation services and AR.



Trial	SDK	Cloud Recognition	Cloud API
€ 0	€ ~~499/one time fee~~ 299 /one time fee	€ ~~133/mo~~ 99 /mo	€ ~~149/month~~ 99 /month
Unlimited local markers	**Unlimited** local markers	**Unlimited** databases	**1,000** calls/mo
1 *demo* app (android, iOS)	**1** app (android, iOS)	**1500** cloud markers	**Unlimited** markers
20 Cloud markers	~~Cloud recognition~~	**1.5 M** Cloud scans / mo	**Unlimited** databases
~~Cloud API~~	~~Cloud API~~	~~Cloud API~~ (optional)	Cloud Recogn. required
~~e-mail support~~	e-mail support	e-mail support	e-mail support

Figure 2.31: Pikkart prices. Reference: [63]

It has its own documentation and a forum where topics about its use are addressed, but its functionalities are paid as displayed in figure 2.31.

2.5.5 Vuforia

Vuforia [97] is an SDK that offers support to build AR applications for mobile devices (Android and iOS) and AR glasses. It can be built in Unity and has compatibility with the AR Foundation SDK.

Vuforia provides the following group of functionalities:

- **Image Targets** - Provides Image recognition and tracking and allows the creation of image databases using the Vuforia Target Manager tool.

- **Multi Targets** - Allows the use of more than one Image Target and arrange them into regular geometric shapes or in any arbitrary arrangement of planar surfaces.

- **Cylinder Targets** - Recognize images wrapped onto objects that are cylindrical or close to in shape (e.g. bottles). It also allows the adjustment of the image size.

- **VuMarks** - VuMarks are customized markers that can encode a range of data formats supporting unique identifications and tracking.

- **Model Targets** - Allows the recognition of objects by shape using pre-existing 3D models and places AR content on items like industrial equipment, vehicles, toys, and home appliances.

- **Object Targets** - Object Targets are created by scanning an object with the Vuforia Object Scanner. It's the best option for toys and other objects with a lot of surface details.

- **Area Targets** - These are environments that can be scanned using the Vuforia Area Target Creator app or other commercially available 3D scanners. This creates persistent content into a wide variety of places.

- **Ground Plane** - Allows the placement of content on horizontal surfaces in the environment like tables and floors.
- **Vuforia Fusion** - Fusion detects the capabilities of the underlying device and fuses them with Vuforia Engine's features. Vuforia Engine can, for that reason, be used alongside the AR Foundation Platform and uses ARCore and ARKit as its foundation (figure 2.32).

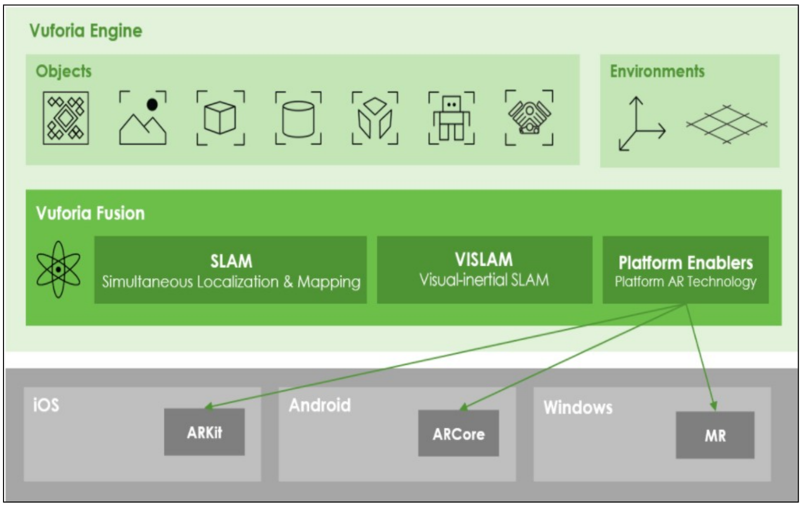
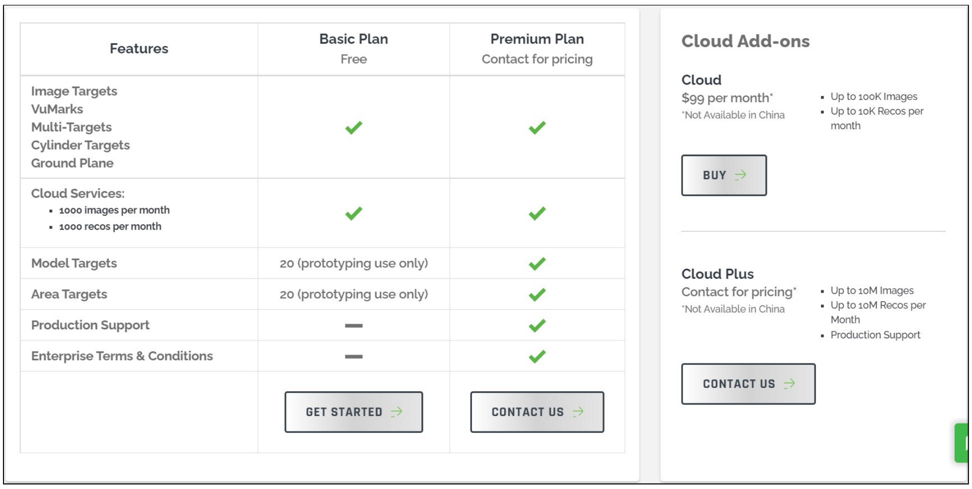


Figure 2.32: Vuforia Fusion Schematic. Reference: [98]

- **Device Tracking** - Vuforia Engine offers a tracking functionality by using the Device Pose Observer. This allows tracking a Vuforia target when the object or content is no longer in the camera view.
- **Cloud Recognition Service** - Recognize a large set of images and frequently update the database with new images.
- **Data and Driver Framework** - Allows developers to provide and consume data from external systems through Vuforia Engine and to record sessions and create custom drivers for their devices.



Features	Basic Plan Free	Premium Plan Contact for pricing
Image Targets VuMarks Multi-Targets Cylinder Targets Ground Plane	✓	✓
Cloud Services: • 1000 images per month • 1000 recos per month	✓	✓
Model Targets	20 (prototyping use only)	✓
Area Targets	20 (prototyping use only)	✓
Production Support	—	✓
Enterprise Terms & Conditions	—	✓
	GET STARTED →	CONTACT US →

Cloud Add-ons

Cloud
$99 per month*
*Not Available in China
- Up to 100K Images
- Up to 10K Recos per month

BUY →

Cloud Plus
Contact for pricing*
*Not Available in China
- Up to 10M Images
- Up to 10M Recos per Month
- Production Support

CONTACT US →

Figure 2.33: Vuforia Prices. Reference: [99]

Vuforia offers support to most devices with Android version 6+ or iOS 11+. Some functionalities like VISLAM or compatibility with ARCore are only specific to certain devices. Vuforia has its own portal and forums where access to documentation is provided. Despite having a basic plan for free, this one doesn't offer many functionalities as shown in figure 2.33.

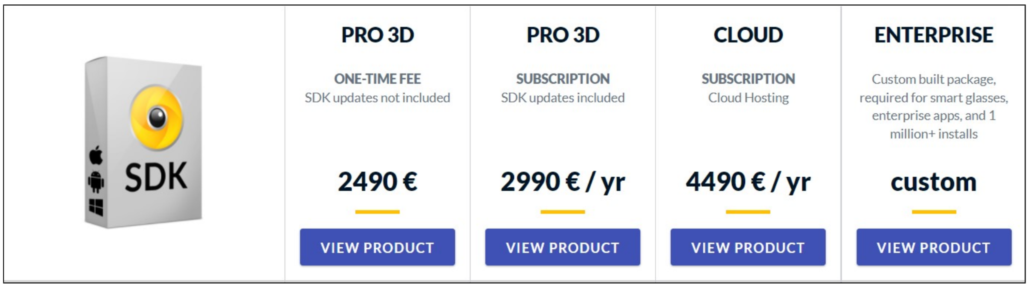
2.5.6 Wikitude

Wikitude AR SDK [104] is another alternative to the previous SDK's. It offers different marker-based and markerless AR functionalities:

- **Image Tracking** - Image Tracking allows AR apps to detect, track, and augment 2D images. The Multiple Target functionality enables the tracking of more than one image at a time.
- **Object and Scene Tracking** - Just like Image Tracking recognizes images and planar surfaces, Object Tracking detects 3D structures and objects based on an object's pre-recorded map.
- **Instant Tracking** - Instant Tracking uses Wikitude's Simultaneous Localization and Mapping (SLAM) technology. SLAM identifies the user's location within an environment by simultaneously mapping the area during the Instant Tracking AR experience. This allows environment mapping and AR content display without markers.
- **Multiple Image Targets** - Multiple Image Targets enables AR apps to track and augment several images simultaneously.
- **Cloud Recognition** - Cloud Recognition is the online recognition solution for large-scale AR projects and allows recognition of up to 1,000 images without a network connection.
- **Geo AR** -AR development framework that allows creating geo-referenced data.
- **Cylinder Tracking** - Cylinder Tracking allows AR apps to detect, track and augment cylinder targets. The Multiple Cylinder Tracking functionality enables tracking of more than one cylinder target at a time.
- **Multiple Object Tracking** - Multiple Object Tracking enables AR apps to track and augment several 3D objects simultaneously.
- **Multiple Trackers** - Makes it possible to combine multiple AR features simultaneously.

Wikitude offers development support in different technologies such as Flutter, Cordova or Xamarin through a Javascript API. It also provides support to Unity. It's used for development of smart glasses applications as well as smartphone applications (Android 6.0+ or iOS 12.0+ required).

Wikitude's official website provides documentation, tutorials, links to developer blogs and to its forums. However, its functionalities aren't available for free as we can see in figure 2.34.



SDK	PRO 3D	PRO 3D	CLOUD	ENTERPRISE
	ONE-TIME FEE SDK updates not included	SUBSCRIPTION SDK updates included	SUBSCRIPTION Cloud Hosting	Custom built package, required for smart glasses, enterprise apps, and 1 million+ installs
	2490 €	2990 € / yr	4490 € / yr	custom
	VIEW PRODUCT	VIEW PRODUCT	VIEW PRODUCT	VIEW PRODUCT

Figure 2.34: Wikitude Prices. Reference: [105]

2.5.7 Location Pluggins

Since this project will focus on the location of the user's device, there was the need to search for available tools and technologies on device's capabilities related to tracking. Most of the existing solutions in the market for indoor navigation rely on Bluetooth low energy (BLE) beacons or markers. It was decided that the best solution would be to develop a prototype relying on markers, that, if necessary would be using beacons to track its location with more precision. Nevertheless the following solutions were also considered:

Matterport with Vuforia Area Target Generator

Vuforia Area Targets are created from a 3D scanned digital model of a physical space. Such a model can be obtained by using a 3D depth scanning camera. Currently, the Matterport Pro2 Camera and the Leica BLK360 is supported for this feature [96].

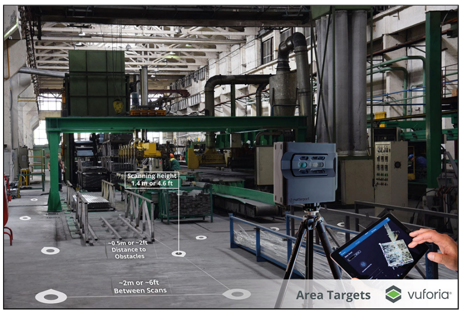


Figure 2.35: Matterport with Wikitude scan. Reference: [96]

The Matterport Capture App is required to control the Matterport camera and is only supported on iOS devices. The app connects to the Matterport camera to control and manages the scanning process as shown in figure 2.35. It is free to download from the Apple App Store.

Cloud Anchors

Cloud Anchors [40] are similar in behavior to anchors but they're hosted on a cloud service allowing users to interact with these objects simultaneously from different positions (figure 2.36). However there are several disadvantages, since it takes significant time to load (10 seconds to 2 minutes) and they're not tolerant to environment changes

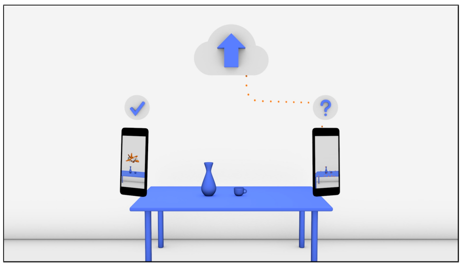


Figure 2.36: Google Cloud Anchor. Reference: [51]

2.5.8 Analysis and Conclusion

After discussing all the available options with all interested parties in the project, it was decided that AR Foundation would be the best SDK available since it's free, has good documentation as well as available repositories, and it's compatible with the existing system. Relatively to the location, it was decided that it should be developed an indoors tracking functionality in the existing mobile application. Bellow there's table 2.1, where a comparison between all SDK's is made. The community size or popularity of each SDK is evaluated based on the number of repositories and commits. Its documentation is evaluated based on the author observations and opinion where a score between 1 and 5 is given.

SDK Feature	ARCore	ARKit	ARF.	Pikkart	Vuforia	Wiki.
Free	Yes	Yes	Yes	No	No	No
Documentation	5	4	5	3	5	4
Android Support	Yes	No	Yes	Yes	Yes	Yes
iOS Support	No	Yes	Yes	Yes	Yes	Yes
Unity Support	Yes	Yes	Yes	Yes	Yes	Yes
Marker-based AR	Only 2D	Yes	Yes	Yes	Yes	Yes
Markerless AR	Yes	Yes	Yes	Yes	Yes	Yes
Location-based AR	No	Yes	Yes	Yes	No	Yes
Number of repositories	2K	4K	0.5K	22	2K	0.3K
Number of commits	9K	7K	0.9K	9	10K	1K

Table 2.1: SDK comparison.

Chapter 3

System Description

This chapter will present all the functional and non functional requirements, as well as providing a better context on the scope of the project through use cases and constraints. It will also describe the system's architecture and some of its functionalities.

3.1 Concepts

In order to better understand this section, there are some concepts that need to be explained in advance.

- **Building** - Represents a building containing information associated with it.
- **Point of Reference** - Represents a location to start to navigate a building or to correct the mobile users position through a QR Code.
- **Point of Interest** - Represents an important location and potential destination within the building.
- **Exhibition** - Represents a group of objects that are displayed within a time frame in a certain building.
- **Exhibit Object** - Represents an object to be displayed in an exhibition.
- **Exhibit Component** - Represents an exhibit object but with its location within a building.
- **Signboard** - Represents a board displaying a collection of signs to direct users.
- **Sign** - Represents an object that gives information and directions.

In addition, it is necessary to distinguish the different users according to their access permissions to the system.

- **Content Manager** - User able to perform every action available through Back-end services.

- **Mobile Application User** - User with access to the mobile application that provides AR experiences.

3.2 Use Cases

Use cases [29] are used in system analysis to identify, clarify and organize system requirements. A use case describes an interaction between the system and an actor to reach a certain goal. The following format was used to represent the project's use cases:

- **Name**: An identifier that represents the purpose of the use case.
- **Actor**: Agent involved in the use case.
- **Description**: Goal to be achieved in the use case.
- **Pre-Conditions**: State of the system before the use case scenario happens.
- **Basic Flow**: Steps that the actor follows to make sure that the purpose of the use case is met.
- **Post-Conditions**: State of the system after the use case scenario happens
- **Alternate Course**: Represents alternative or undesirable paths to the actor.

For this project, the use cases are divided into two sections, one for the Mobile application use cases and another for the Back-end uses cases. The list of all of the use cases is in Appendix A. From those, the following tables (3.1 and 3.2) were considered the most relevant:

Name	Choose point of interest to navigate through the building (FR-27)
Actor	Mobile Application User
Description	The mobile app user can choose a point of interest to get assistance while navigating towards it.
Pre-conditions	The mobile application content includes the building required information.
Basic Flow	1. User navigates through the mobile application until reaching the navigation state. 2. User chooses one of the available options from the points of interest's list. 3. Mobile Application shows the path to the selected location. 4. User reaches point of interest location. 5. Mobile Application shows 3D model associated with the selected point of interest.
Post-conditions	User gets to the destination and is allowed to exit the navigation screen.
Alternate Course	3a. Mobile application does not recognize current location due to drifting. 3a.1. Mobile Application user points the camera towards a point of reference. 3a.2. Mobile Application recognizes point of reference and corrects its current position (jump to 3.).

Table 3.1: Choose point of interest to navigate through the building.

Name	Scan point of reference (FR-26)
Actor	Mobile App User
Description	The mobile app user scans a point of reference to get into the navigation screen or to correct its position.
Pre-conditions	The mobile application content includes the building required information.
Basic Flow	1. Mobile Application user navigates through the mobile application until reaching the camera state. 2. Mobile Application user points the camera towards a point of reference in the building. 3. Mobile Application recognizes point of reference. 4. Mobile Application opens the navigation screen with its current position.
Post-conditions	Mobile Application opens the navigation screen with the device's current position.
Alternate Course	3a. Mobile application does not recognize current location due to drifting. 3a.1. Mobile Application user points the camera towards a point of reference. 3a.2. Mobile Application recognizes point of reference and corrects its current position (jump to 4.).

Table 3.2: Scan point of reference.

3.3 Functional Requirements

Functional requirements define functions or features of a system. A functional requirement may have a unique name and number, a description, and a priority associated. All of these fields are necessary so that its easier to understand its need, its goal and to track the requirement during development [37]. The functional requirements were prioritized in accordance with their importance for the project. For this prioritization, an adaptation of the MoSCoW prioritization method [25] was used, which was divided in:

- **Must Have** - Requirement is crucial for the project. Cannot deliver a viable solution without it.
- **Should Have** - Requirement is important but not vital.
- **Could Have** - Requirement is wanted or desirable but less important.

From the list of requirements produced for this project in Appendix B, the ones that are associated with the most important use cases are in table 3.3.

ID	Name	Description	Priority
FR-25	Update Building's App Content	The Mobile Application must be able to update its building's related content.	Must Have
FR-26	Read QR Codes	The Mobile Application must be able to read and interpret QR Codes.	Must Have
FR-27	Show Shortest Path	The Mobile Application must be able to calculate and show the shortest path to a destination.	Must Have
FR-28	Show Building's Content	The Mobile Application must be able to show 3D models in designated locations within the building.	Must Have

Table 3.3: Mobile Application Functional requirements with high priority.

3.4 Non-Functional Requirements

While functional requirements describe functions that a software should perform, non-functional requirements define the quality attribute of a software system and how it should work [1]. All of these are essential for a system to work correctly and therefore are prioritized as a Must Have.

3.4.1 Performance

It is important to ensure that the application does not have performance issues on all devices on which it is available. To ensure this, the way the application works is modified accordingly and the navigation system does not run in parallel with the content of the previous system. The number of polygons present in a 3D model may impact the Back-end and the Mobile application's performance, as some tests and its consequent decisions were taken as follows:

- **Mobile Application**: The greater the number of polygons present in a 3D model's mesh the greater the time needed to its rendering process. This process heavily depends on hardware capabilities. For this reason, the mobile application frame rate was tested under different conditions and a specific number of fifteen frames per second was considered to be its acceptance value. As a result, there's a limit to the number of objects per exhibition and to the size of each associated file.

- **Back-end**: The necessary time to process buildings, points of interest and exhibit objects was also tested, having an acceptance value of 10 minutes. Tests were performed to objects that would fit the above conditions and none of them took more time than the acceptable value to process. It was then considered that the file size limit previously imposed was enough as a measure to guarantee performance in the Back-end.

Further details on these tests are given in the Development and Testing chapter.

3.4.2 Robustness

If the mobile application cannot connect to the server, it will have to fall back on the locally stored content without this having any negative impact on its operation. Furthermore, if drifting occurs during the navigation experience, the mobile application can reposition itself through scene cloud points tracking or reading points of reference's QR Codes.

3.4.3 Security

The system ensures that only content manager users have access to the Back-end private API functionalities. In addition, it only allows connections from mobile application users to the public API.

3.5 Constraints

Technical constraints are specifications that limit the design choices of the project planning [93]. The constraints for this project are displayed in tables 3.4, 3.5 and 3.6.

ID	C-1
Source	Critical
Title	Back-end in .Net Core 3.1
Description	Since the previous system has a Back-end in the framework .Net Core 3.1, any changes or additions to it will be done using the same framework.

Table 3.4: Constraint 1 - Back-end in .Net Core

ID	C-2
Source	Critical
Title	Microsoft Azure VM
Description	The current system Back-end is hosted in a Microsoft Azure Virtual Machine, thus the Back-end will be hosted on the same service.

Table 3.5: Constraint 2 - Microsoft Azure hosting

ID	C-3
Source	Critical
Title	Mobile Application in Unity
Description	The existing mobile application was developed using Unity3D, thus any changes or additions will be done using the same engine.

Table 3.6: Constraint 3 - Mobile Application in Unity

3.6 Architecture and Functionalities

When a system's architecture is designed, it is possible to understand the structure and organization of the system and the relationships between its components [92]. A representation of the developed system(ARCritical version 2.0) architecture is featured in figure 3.1.

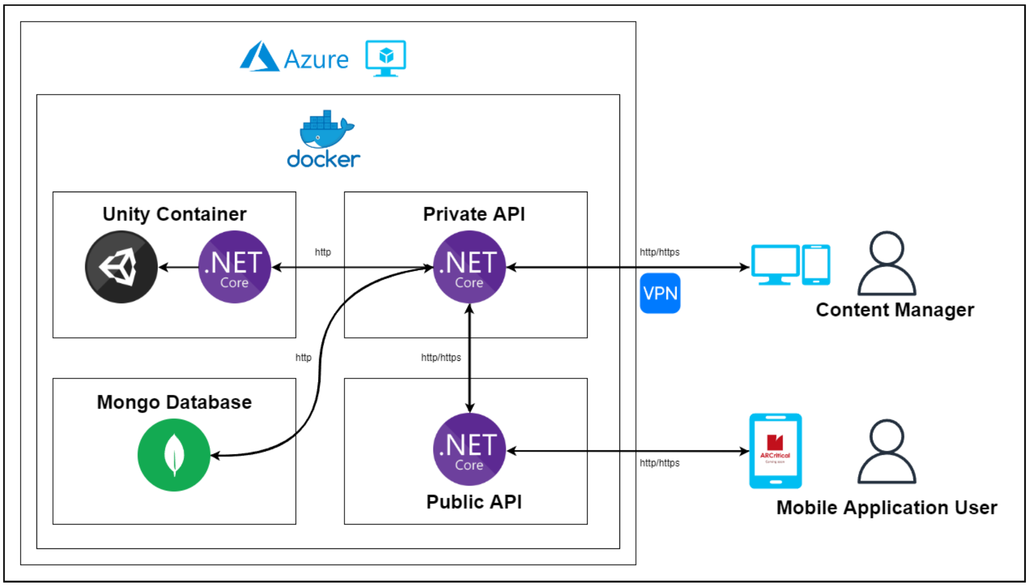


Figure 3.1: Architecture of ARCritical version 2.0.

3.6.1 Mobile application

Unity is usually used for game development however its capabilities allows its use to special effects and virtual and augmented reality applications. This mobile application was developed in C# using the Unity engine having three different scenes:

- **Startup Scene**: The first screen that allows the user to update the mobile application content.
- **AR Tracking Scene**: The second screen that provides the previously existing functionalities related with image tracking.

- **Navigation Scene**: The third screen that provides indoors navigation assistance and allows the user to experience an exhibition of 3D digital objects.

Further down in this section, there are more details on the changes made in the first two previously existing scenes and on the Navigation Scene.

Startup Scene

In this scene, one of the most important aspects is the fact that it is the first screen presented to the user and allows him to update the mobile application content as shown in figure 3.2.

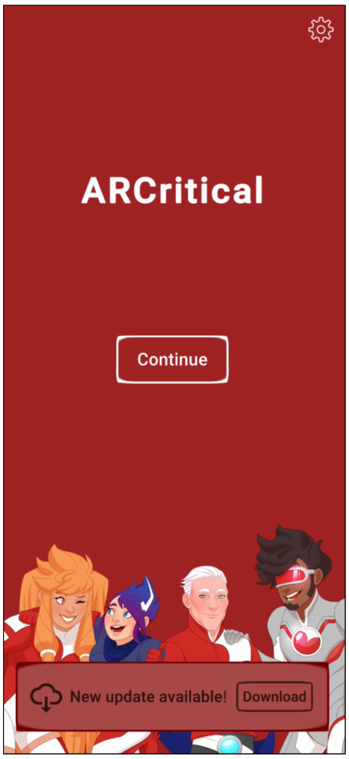


Figure 3.2: Startup Scene screen view.

The mobile application updates the content for it to be used during the AR Tracking Scene. To accommodate the necessary changes mentioned in the functional requirement RF-25, some additions were necessary. The following steps explain how this requirement is met:

- **1°**: The mobile application loads any locally stored information of the buildings.
- **2°**: The mobile application requests the back-end for information of all the buildings, including its nested objects (exhibitions, points of interest, points of reference and signboards).

- **3º**: The mobile application checks if there's any update to be made, comparing to what's stored in cache, sending requests to the back-end to download any new content(3D models for Android or iOS).

- **4º**: The mobile application deletes outdated content from the cache.

All of the content - 3D models of buildings, points of interest, and exhibit objects - needs to be processed and updated to be loaded in run-time, thus, avoiding compiling and building the application. This is possible using AssetBundles, which are bundles of different types of content (3D models, textures, audio files) that can be created using the Lempel–Ziv–Markov chain algorithm (LZMA) and cached with the LZ4 compression algorithm [86]. This brings even more advantages due to lower memory consumption, lower download time and lower space required to be stored in cache.

AR Tracking Scene

This scene employs the already existing functionalities. It makes use of the following AR Foundation supported features [85]:

- **Device Tracking**: tracks the device's position and orientation in the physical space.

- **Plane and Feature Point Detection**: detects horizontal and vertical surfaces.

- **Image Tracking**: detects and tracks 2D images.

- **Anchors**: tracks an arbitrary position and orientation.

The figure 3.3 features this scene, while an image gets recognized and a 3D model is placed on top of it.

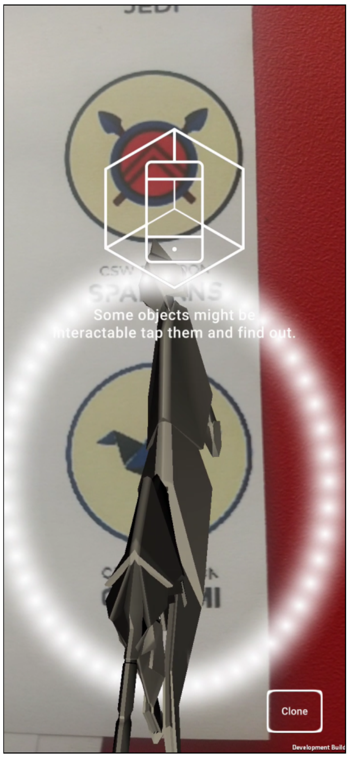

Some objects might be interactable tap them and find out.

Figure 3.3: AR Tracking Scene screen view.

To allow a change between scenes some additions were necessary. It was decided that the best way to seamlessly change to the Navigation Scene would be to read a point of reference's qr code as described in the functional requirement FR-26. This was the only change that was required in this scene.

Navigation Scene

This is the scene where the majority of the mobile application functional requirements were implemented. The following steps of how the application starts the scene shows how these requirements are met:

- **1º**: The mobile application loads the 3D model of the building.
- **2º**: The mobile application loads the points of interest's list and its 3D models.
- **3º**: The mobile application loads the rest of the content including exhibit object's 3D models and signboards.
- **4º**: The mobile application displays a drop-down list with points of interest, an expandable mini-map and a back button.

The mobile application allows the user to chose a point of interest of the drop-down list, showing the shortest path to it as shown in figure 3.4 (FR-27).

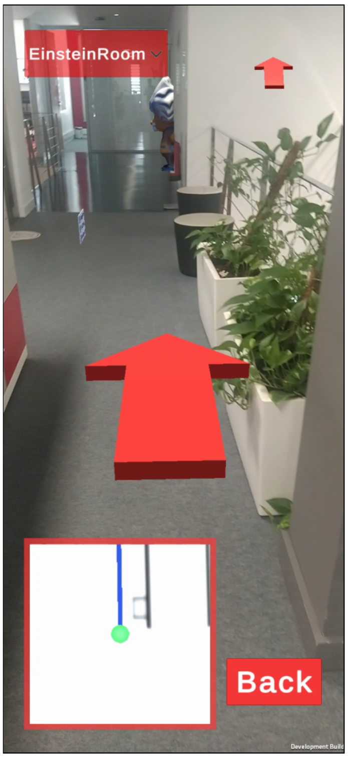


Figure 3.4: Navigation Scene screen view with a point of interest selected.

The mobile application has also the feature to expand the map of the building to full screen (FR-29) as in figure 3.5.

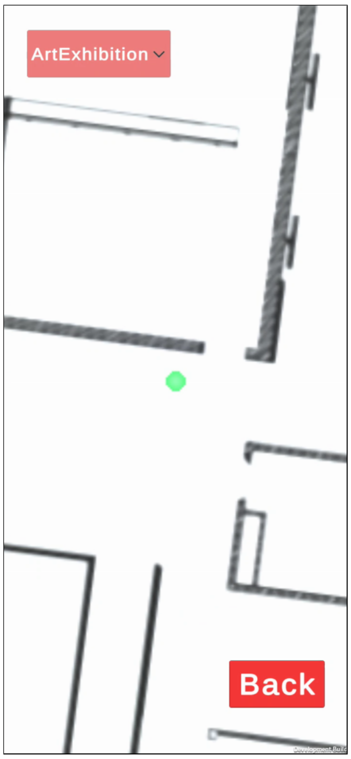


Figure 3.5: Navigation Scene screen view with expanded map.

While moving through the building, all of its associated content (signboards or exhibit objects) is displayed on the screen, as featured in figure 3.6, allowing the user to have an AR experience based on its location (FR-28).

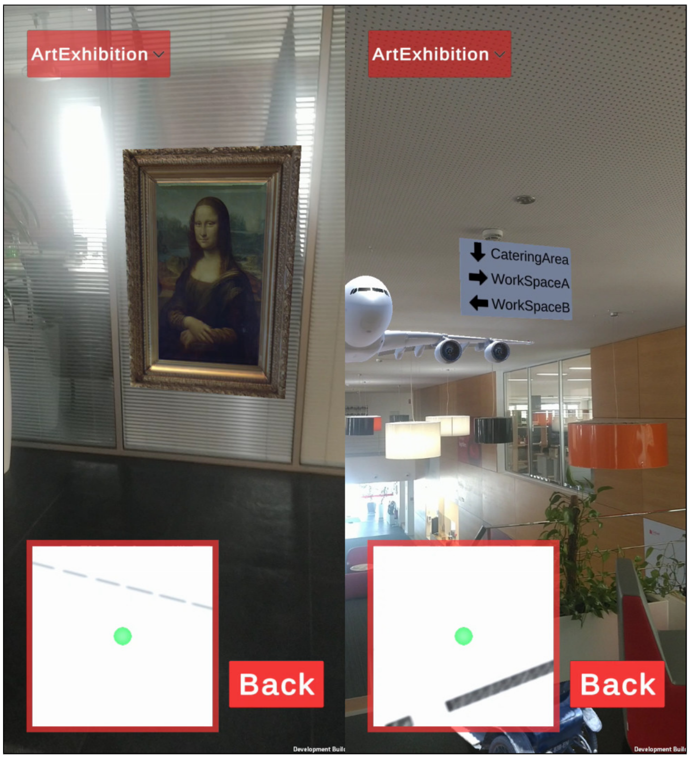


Figure 3.6: Navigation Scene screen views with Exhibit Objects and Signs.

In addition to this, the Navigation Scene also fulfills the requirement FR-26, since it reads points of reference's qr codes to reposition the device in the virtual environment. Further details on how some of these features were accomplished are further described in the development chapter.

3.6.2 Back-end

All of the Back-end runs on a docker instance with several containers, hosted on a virtual machine from Microsoft Azure services. There are 4 main containers:

- **Private API**: This is the main container that provides most of the necessary functionalities defined in the functional requirements.
- **Public API**: This container works as a Proxy to the Private API as it exposes some of its functionalities for the mobile application.
- **Unity Container**: This container processes all of the 3D models, its textures and associated AssetBundles.
- **Database**: This container stores the necessary data.

Private API

The Private API is the most important container as it is responsible for providing most of the functionalities described in the functional requirements tables B.1 B.2 and B.3. It was developed using the framework .Net Core 3.1 based on an Onion Architecture. The layers of this design are separated to maximize flexibility and reduce coupling as we can see in figure 3.7.

- **Presentation**: layer that provides the access to endpoints.
- **Application**: layer retaining the application logic and the Data Transfer Objects (DTO).
- **Domain**: layer that represents the business objects and interfaces.
- **Data**: layer that access the data.
- **Infrastructure**: outermost layer holding configurations of the other layers.

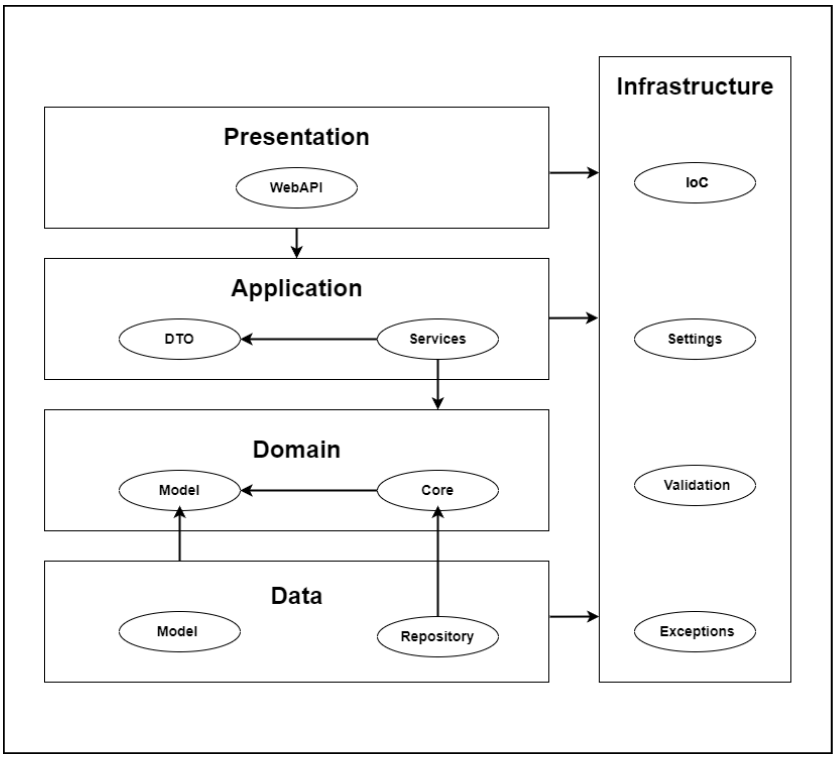


Figure 3.7: API Layers based on the Onion Architecture.

The Private API container also uses QR Coder (version 1.4.3)[30], to generate the QR Code image files as Points of Reference are added.

Public API

As previously mentioned, the Public API serves the purpose of a proxy for the Private API, providing only the required methods from the Mobile Application.

The Public API provides a total of twelve Private API endpoints,seven of which are part of the existing system. From the new endpoints, one returns all the buildings, while the others allow the download of AssetBundles of buildings, points of interest and exhibit objects for Android and iOS platforms.

Unity Container

This container processes all of the 3D models, its textures and associated AssetBundles to allow them to be loaded during run-time in the mobile application. As the Private API receives new content (buildings, points of interest or exhibit objects), it sends a request to this container's API and one process of the Unity Editor is created to make an AssetBundle for each of the platforms of Android and iOS. After both AssetBundles are successfully created, a response is sent to the Private API which updates its AssetBundle status in the database.

Database

The current system stores data in Json/Bson documents utilizing NoSQL using MongoDB. This makes it easier to convert the Private API calls into CRUD operations (Create, Read, Update, Delete), as well as to manipulate the data structures in the event that modifications are needed. For these reasons, it was decided that maintaining the technology in use with a data structure based on nested objects would be the most appropriate. This database has two collections:

- **Buildings**:
  - **Name**: Name of the building.
  - **File Name**: File name of the building's 3D model.
  - **File Extension**: File extension of the building's 3D model.
  - **AssetBundle Status**:
    - Value -1: There was an error processing the building's AssetBundle.
    - Value 0: The building's AssetBundle is currently being processed.
    - Value 1: The building's AssetBundle was successfully processed.
  - **Points of Interest**: Nested objects consisting of points of interest.
  - **Points of Reference**: Nested objects consisting of points of reference.
  - **Signboards**: Nested objects consisting of signboards.
  - **Exhibitions**: Nested objects consisting of exhibitions.
- **Exhibit Objects**:
  - **Name**: Name of the exhibit object.
  - **File Name**: File name of the exhibit object's 3D model.
  - **File Extension**: File extension of the exhibit object's 3D model.

    - **AssetBundle Status**:
        - Value -1: There was an error processing the exhibit object's AssetBundle.
        - Value 0: The exhibit object's AssetBundle is currently being processed.
        - Value 1: The exhibit object's AssetBundle was successfully processed.

In addition to these two collections, these are the nested objects inside the buildings collection:

- **Point of Reference**:
    - **File Name**: File name of the generated QR Code file.
    - **File Extension**: File extension of the generated QR Code file.
    - **Position**: Coordinates of its location inside the building.
    - **Rotation**: Rotation inside the building.
- **Point of Interest**:
    - **Name**: Name of the point of interest.
    - **File Name**: File name of the point of interest's 3D model.
    - **File Extension**: File extension of the point of interest's 3D model.
    - **Position**: Coordinates of its location inside the building.
    - **Rotation**: Rotation inside the building.
    - **AssetBundle Status**:
        - Value -1: There was an error processing the point of interest's AssetBundle.
        - Value 0: The point of interest's AssetBundle is currently being processed.
        - Value 1: The point of interest's AssetBundle was successfully processed.
- **Signboard**:
    - **Position**: Coordinates of its location inside the building.
    - **Rotation**: Rotation inside the building.
    - **Signs**: Nested objects consisting of signs.
- **Sign**:
    - **Name**: Name of the sign.
    - **Direction**: Direction to point towards.
- **Exhibition**:

  - **Name**: Name of the exhibition.
  - **Starting Date**: Date of the start of the exhibition.
  - **Ending Date**: Date of the end of the exhibition.
  - **Exhibit Components**: Nested objects consisting of exhibit components.

- **Exhibit Component**:
  - **Position**: Coordinates of its location inside the building.
  - **Rotation**: Rotation inside the building.
  - **Exhibit Object**: Reference to an Exhibit Object.

Chapter 4

Project Planning

This chapter will present the chosen development methodology, the tools being used in the project, the task planning using a Gantt chart and finally the risk management.

4.1 Methodology

Since this project requires a high degree of freedom in its development, not only because it is a recent technology that may require changes but also because feedback received may require changes, it was decided that an agile approach would be the most appropriate.

Agile is a structured, iterative approach to project management and product development. It provides a way for self-organizing teams to respond to change addressing the inherent unpredictability of product development [13].

Scrum and Kanban were the two Agile methodologies to be considered since their characteristics proved to be the most adequate for this project's development needs [17].

Scrum is an agile development methodology used to develop software based on an iterative process. Scrum is an agile framework designed to provide value to the customer throughout development and to allow changes at the end of each sprint to adapt the project to the feedback received [19; 78].

Kanban is an agile development methodology used in software development that focuses on workflow management by task. In this method, there is a set of tasks that can be managed through a Kanban board. This board contains three columns where tasks are arranged according to their status: To Do, In Progress, and Done. This creates a flow that avoids bottlenecks and allows changes to the board during development [18; 58].

The chosen methodology for this project is the **Agile Kanban**. Since the development team for this project consists only of the intern, it is not necessary to introduce sprints as it's easier to manage the project with a Kanban board.

4.2 Tools

Having decided on what technologies this project would rely on, it's important to mention the support tools employed throughout the different stages of the project:

- **Unity Editor** - Platform used during development of the mobile application [90].
- **Visual Studio** - IDE and code editor used during development [54].
- **Confluence** - Platform where any documentation or knowledge related to the project are stored [15].
- **Jira** - Platform where the backlog and the requirements are registered. Allows management of the tasks providing a Kanban board [16].
- **BitBucket** - Tool that provides repositories to store the artifacts related to the project and to do version control [14].
- **Overleaf** - Online LaTeX editor used to produce this project's document [59].
- **Docker** - To integrate the code in a shared environment and deploy more efficiently between environments [33].
- **Visual Paradigm** - Used to produce the necessary charts related with the planning of the project [95].
- **Microsoft Teams** - Messaging platform used for communication [53].
- **TeamGantt** - Online Gantt chart maker [79].
- **Blender** - Free and open-source 3D computer graphics software tool set used for creating 3D models and virtual reality content [23].
- **WinSCP** - SFTP client and FTP client for Microsoft Windows to establish connections between a local computer and remote servers [106].
- **DWG TrueView** - Software to view DWG files or convert them to work with older versions of AutoCAD software [20].
- **Robot T3** - Free, lightweight, open-source MongoDB GUI with an embedded mongo shell [69].

4.3 Planning

In this section, the planning of the project will be discussed and presented through Gantt charts, as well as the necessary changes made during the internship.

4.3.1 First Semester

Regarding the project's first semester planning (figure 4.1), there were some setbacks due to personal circumstances that delayed a couple of the planned activities as shown in figure 4.2. However, most of the plan's tasks were successfully accomplished within this semester with the exception of the prototype.

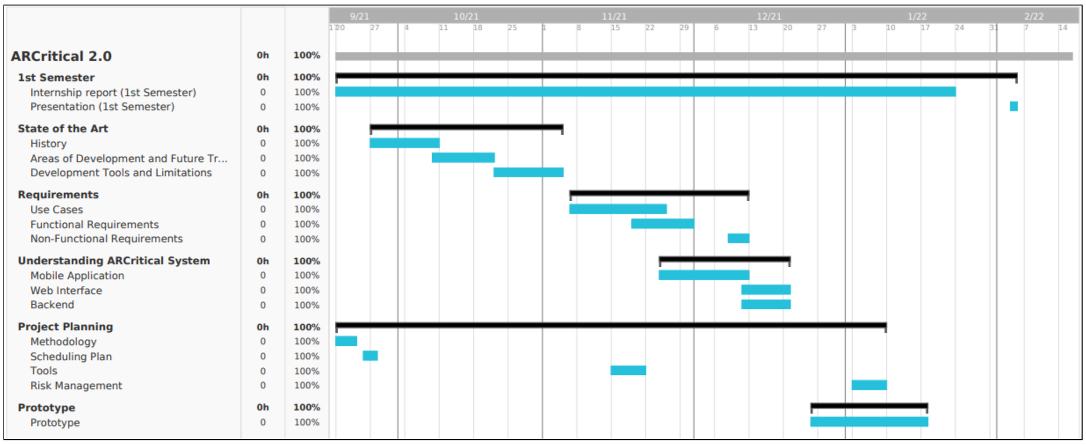


Figure 4.1: Gantt Chart: first semester plans.

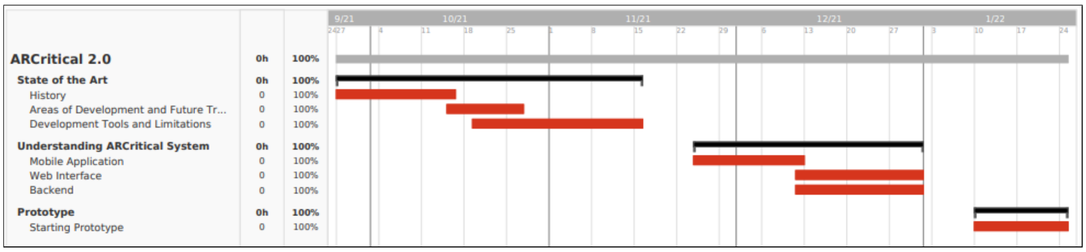


Figure 4.2: Gantt Chart: first semester adjustments.

In the first meeting with the Critical team, it was decided that the first task would be to start the State of the Art on AR while brainstorming potential ideas that could enrich the internship. This provided some freedom to the author, allowing him to have a say on some of the decisions within the future project. During this stage, the author researched on similar projects, cross-platform technologies and the history of AR. These tasks would extend for the first seven weeks of the semester.

Following the State of the Art, the System Description stage would consist of the next six weeks. This stage would start with the description of the use cases based on user stories discussed in scheduled meetings. Following the use cases, the list of functional and non-functional requirements was produced, as well as, its priorities. The system's architecture was also discussed and defined, since during this stage some of the time was dedicated to fully understand the already existing system.

As for the Project planning, the methodology and scheduling plan was defined in the first week and the necessary tools were determined during the survey of the requirements. The last week of December's work was towards risk management.

Starting in the first week of January until the delivery of the intermediate report in the end of this month, the effort would be focused towards the writing of the intermediate report, as well as, starting the prototype.

Since there were some delays in the conclusion of The State of the Art and those propagated to the next stages, most of the prototype expected effort had to be pushed to the second semester.

During all of this period, there were scheduled meetings every week where the progress and future tasks were discussed.

4.3.2 Second Semester

The plan for the second semester follows the figure 4.3. For several reasons, further explained, multiple tasks were changed and required adjustments as figure 4.4 shows.

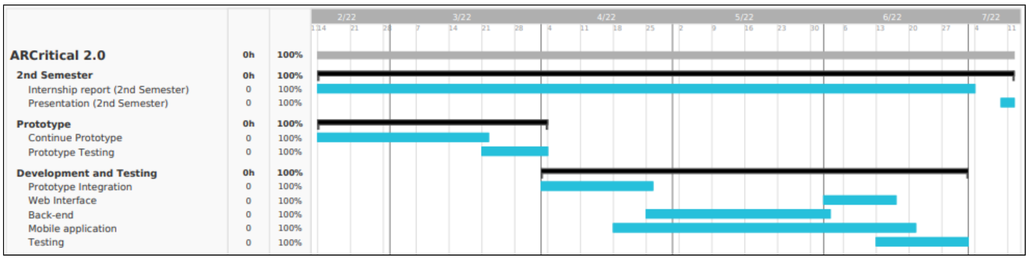


Figure 4.3: Gantt Chart: second semester plans.

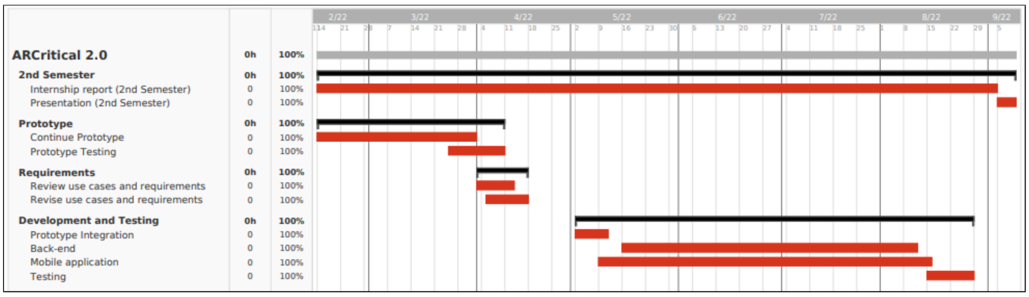


Figure 4.4: Gantt Chart: second semester adjustments.

In the second semester, the Development and Testing stage started with the prototype development in the first seven weeks. During this time it was also necessary to start the modelling process as it was required to later test the prototype in a relevant environment. During the first week of April, the prototype was tested in the Critical's building. As the results of these tests were better than expected, the second week of this month was mostly to review and revise use cases and requirements and start the integration within the Mobile Application.

As shown in figure 4.4, during the last two weeks of April, the author wasn't able to work due to health problems. In the first week of May the integration of the prototype functionalities into the existing mobile application was finished. From the second week of May until the middle of August the Development stage continued with Back-end development and further necessary changes in the Mobile Application. Starting in the third week of August until the delivery of the report, the effort would be focused towards the writing of the report, as well as, conducting some final tests.

4.4 Risk Management

Risk management is the process of identifying, assessing and minimizing the impact of any risk that arises over the project's life cycle [66]. A risk is any event that could impact people, processes, technology and resources of the project. The first step in project risk management is identifying potential risks:

- **R-1** - Incorrect estimation of the task duration and development delays.
- **R-2** - Additional requirements are identified at a later stage.
- **R-3** - Required changes in requirements at a later stage.
- **R-4** - Dependencies between requirements.
- **R-5** - Lack of documentation.
- **R-6** - Lack of experience with some of the technologies.

After identifying the risks, the probability of its occurrence and its impact must be determined. These are divided in three separate categories: high (higher than 75% of probability), medium (25% to 75% of probability) and low (under 25% of probability). The consequences of these risks must then be identified. The last step is to make a mitigation plan to reduce its impact. In table 4.1, the probability, the impact, the consequences and the mitigation plan are presented for each detected risk.

ID	Probability	Impact	Consequences	Mitigation Plan
R-1	Medium	High	It might lead to delays in the delivery of the final product and to drop lower priority requirements.	Estimate and discuss tasks with the advisor.
R-2	Medium	Medium	It might lead to delays in the delivery of the final product and to drop lower priority requirements.	Discuss regularly with the advisor and schedule regular meetings.
R-3	Medium	Medium	It might lead to delays in the delivery of the final product and to drop lower priority requirements.	Discuss regularly with the advisor and schedule regular meetings.
R-4	High	High	It might lead to sudden changes in task management and in development.	Discuss and define tasks' order in the Kanban Board according to its dependencies.
R-5	Low	Medium	Lack of knowledge on the existing system and used technologies.	Create tasks' estimates to allocate time for documentation/system reviews.
R-6	Medium	Low	It might lead to delays in the delivery of the final product and to drop lower priority requirements.	Look through the existing documentation.

Table 4.1: Risk management table.

Chapter 5

Development and Testing

This chapter will provide a summary of the work done within the project by first discussing some of the technologies, decisions and solutions found to fulfill its requirements during development and testing.

5.1 Development

Technology Readiness Level (TRL) is a measurement system that facilitates evaluations of a given technology's maturity [50]. Since the beginning of the internship, it was determined that the main goal was to research, introduce and prove the feasibility of some AR related technologies. The main purpose was to further expand the knowledge on AR, developing and testing an indoors navigation mobile application prototype.

There are some decisions made during this project that are also important to address before going into the development details. Initially, the project's requirements included additions or development of a Web Interface to allow easier access to the new Private API functionalities. Since any content added to a building has to be placed in a determined location with coordinates and rotation values, this process would be done through trial and error. This possibility was thoroughly thought and even with a graphic representation of the 3D model of the building in the web page, it was not considered the best solution to make content management. What was considered to be the best solution was a separate mobile application to accommodate these functionalities, since it would be easier for the Content Manager to place any 3D model inside the building and save its information this way. Since there was not enough remaining time of the internship to make a new mobile application using the Private API, new use cases and requirements were added, and some of the existing ones changed its priority.

Another change made to the requirements of this project is related to the fact that, while testing the device's positional tracking capabilities, the results were better than expected. Initially the requirements didn't take into account that the tracking would be accurate enough while moving between floors. For that reason, the data structure was initially thought to have floors with its respective 3D model associated. Furthermore, there were still some doubts on how much the device would drift under different light conditions and if its position inside the building should be tracked with the help of Bluetooth signal beacons instead of the device's components.

5.1.1 Mobile Application

Prototype

The development of the mobile application started in its prototype. The first step was to check how to track the device's position and rotation in an environment through ARFoundation functionalities. For that, it was needed a 3D model of an office that wouldn't interfere with the camera's AR experience. During the process of modelling, it was also required a VR/SpatialMapping shader to use as material on the walls and floor pieces of the 3D model featured in figure 5.1.

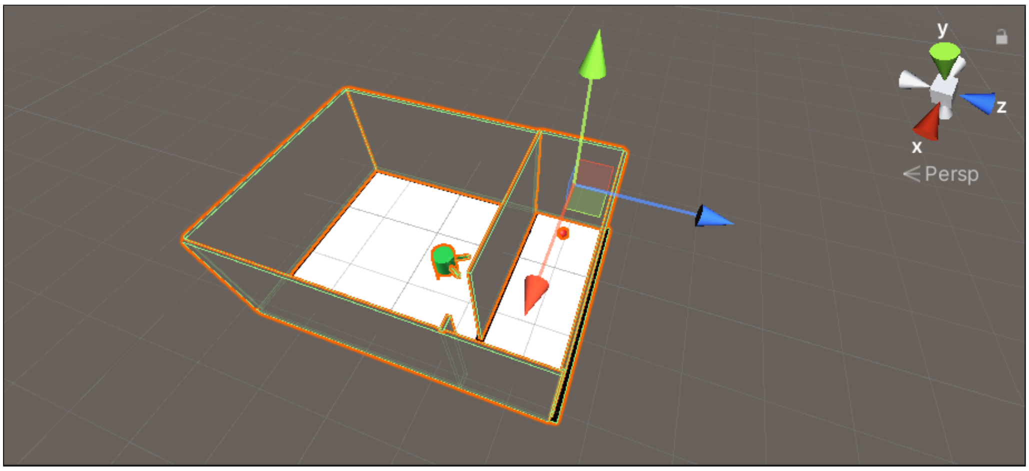


Figure 5.1: 3D model of an office used for the prototype.

It was also necessary, during this prototyping phase, to guarantee that it was possible to capture the AR camera information and process it into a texture, so that it could be later be used to read QR Codes [83].

The library ZXing.Net was then used to read and extract information from the QR Codes if present in the previously mentioned textures [48].

At this point, the prototype in figure 5.2 had been finalized and the next step was to integrate in the existing mobile application.

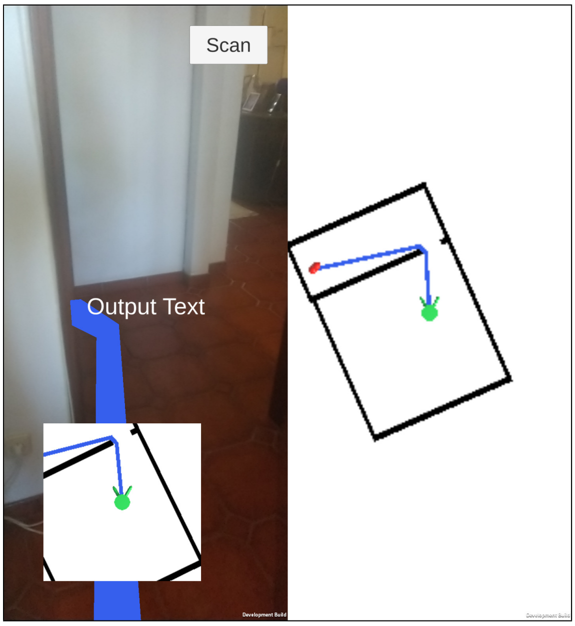


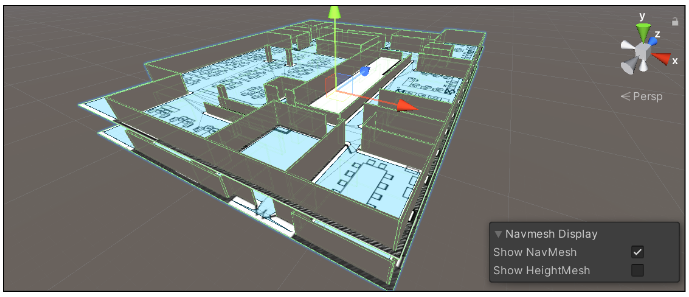
Figure 5.2: Views of the Mobile Application Navigation Scene's prototype.

Integration and Development

To integrate the prototype's functionalities in the existing mobile application and to test them, it was necessary to acquire a 3D model of a Critical Softwares' building. The following steps were taken on the modelling process:

- **Acquire a map for each floor**: DWG TrueView, provided a visualization of a 3D model of the Critical Software's building made in AutoCAD and allowed to extract images of each floor and its measures.
- **Modelling according to measures**: In the Unity3D virtual world, a single unit corresponds to a meter in the real world, which facilitated the process of modelling. Furthermore, it provides a series of tools like Pro Builder [88] that helps in this process.
- **Removing unnecessary data**: Blender was also used to remove unnecessary polygons or vertices from the 3D building model and to change its transform.

The result of this modelling process can be visualized in figure 5.3.



Navmesh Display
Show NavMesh ☑
Show HeightMesh ☐

Figure 5.3: 3D model of a Critical Software building.

This model was then exported to the format fbx (to be imported again in the Back-end) using the fbx exporter functionality in Unity.

As previously mentioned in the architecture, AssetBundles were used to load the content at run-time and to guarantee that the building content could be modified without needing a new build of the application.

In order to get the shortest path to a certain location inside the building, Unity provides the NavMesh functionality. A Navmesh in Unity is a designated mesh which specifies navigable areas in the environment [89]. However, by default, this NavMesh cannot be created during run-time as its needed since the building 3D model is loaded from an AssetBundle sent from the Back-end. For this to happen, a NavMesh Component[87] is created and added to the building floors at run-time and a NavMesh baked after the fact.

These were most of details related to the development process that were considered of most importance. However, much more steps were taken during this process so that it could be possible to reach the final product as in figure 5.4.

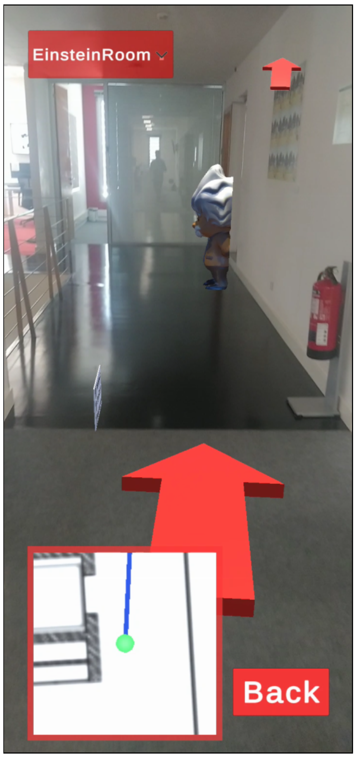


Figure 5.4: Navigation Scene view.

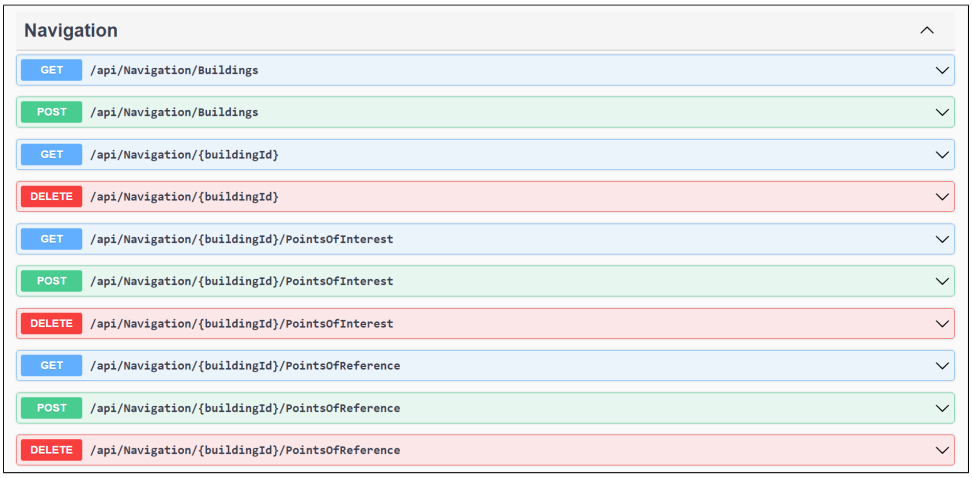
5.1.2 Back-end

The Back-end development was made on an already setup docker environment. The choice of starting development in this environment was made taking into account the advantages that came with it:

- Scalability as many containers can be placed in a single host.

- Fast deployment and ease of creating new instances.
- Ease of maintenance.
- Ease to get support from Critical's personnel.

All of the previously mentioned functional requirements in table B.1. and table B.2. were implemented and are provided in the Private API as we can see in the Swagger UI of figures 5.5, 5.6 and 5.7.



Navigation

GET /api/Navigation/Buildings

POST /api/Navigation/Buildings

GET /api/Navigation/{buildingId}

DELETE /api/Navigation/{buildingId}

GET /api/Navigation/{buildingId}/PointsOfInterest

POST /api/Navigation/{buildingId}/PointsOfInterest

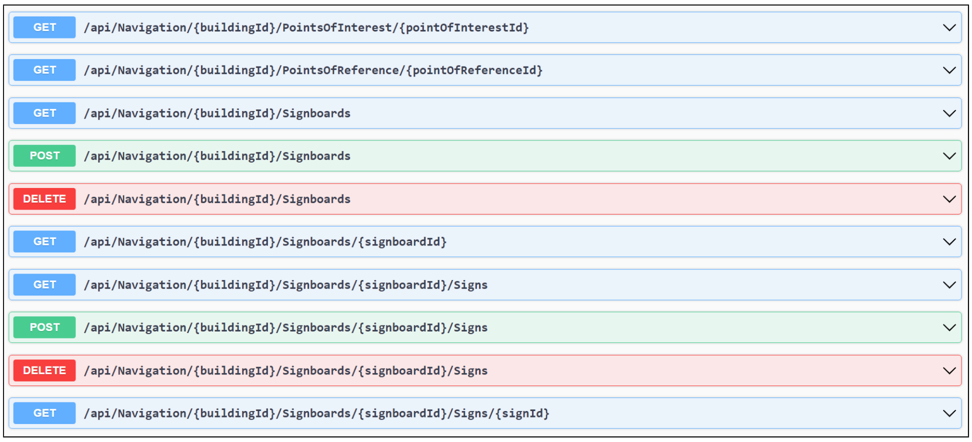
DELETE /api/Navigation/{buildingId}/PointsOfInterest

GET /api/Navigation/{buildingId}/PointsOfReference

POST /api/Navigation/{buildingId}/PointsOfReference

DELETE /api/Navigation/{buildingId}/PointsOfReference

Figure 5.5: Swagger UI preview of some of the defined functionalities.



GET /api/Navigation/{buildingId}/PointsOfInterest/{pointOfInterestId}

GET /api/Navigation/{buildingId}/PointsOfReference/{pointOfReferenceId}

GET /api/Navigation/{buildingId}/Signboards

POST /api/Navigation/{buildingId}/Signboards

DELETE /api/Navigation/{buildingId}/Signboards

GET /api/Navigation/{buildingId}/Signboards/{signboardId}

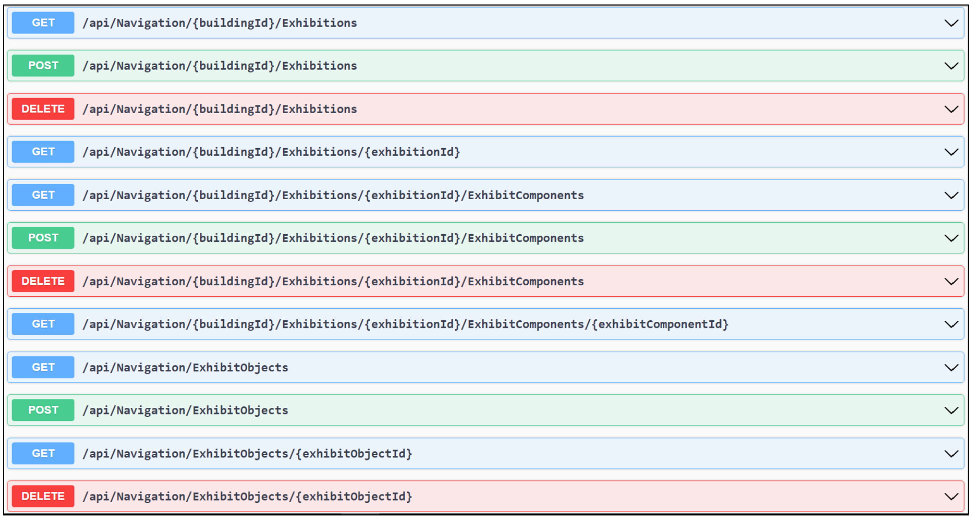
GET /api/Navigation/{buildingId}/Signboards/{signboardId}/Signs

POST /api/Navigation/{buildingId}/Signboards/{signboardId}/Signs

DELETE /api/Navigation/{buildingId}/Signboards/{signboardId}/Signs

GET /api/Navigation/{buildingId}/Signboards/{signboardId}/Signs/{signId}

Figure 5.6: Swagger UI preview of some of the defined functionalities.



```
GET /api/Navigation/{buildingId}/Exhibitions
POST /api/Navigation/{buildingId}/Exhibitions
DELETE /api/Navigation/{buildingId}/Exhibitions
GET /api/Navigation/{buildingId}/Exhibitions/{exhibitionId}
GET /api/Navigation/{buildingId}/Exhibitions/{exhibitionId}/ExhibitComponents
POST /api/Navigation/{buildingId}/Exhibitions/{exhibitionId}/ExhibitComponents
DELETE /api/Navigation/{buildingId}/Exhibitions/{exhibitionId}/ExhibitComponents
GET /api/Navigation/{buildingId}/Exhibitions/{exhibitionId}/ExhibitComponents/{exhibitComponentId}
GET /api/Navigation/ExhibitObjects
POST /api/Navigation/ExhibitObjects
GET /api/Navigation/ExhibitObjects/{exhibitObjectId}
DELETE /api/Navigation/ExhibitObjects/{exhibitObjectId}
```

Figure 5.7: Swagger UI preview of some of the defined functionalities.

During the process of development of these functionalities, some security measures were also implemented like string validations through regular expressions, validation of file formats and validation of file sizes. The requirement FR-6, to create a point of reference, was accomplished by generating QR Codes' image files like the one in figure 5.8 (using QR Coder 1.4.3).

Figure 5.8: Example of generated point of reference's QR Code.

As for the functionalities added in the Back-end Unity container, these were made according to the references of the Unity Scripting API [91] to allow the import of 3D models and applying its textures to be sent in the AssetBundle.

5.2 Testing

This section addresses some of the conducted tests done to the system and how its results changed the requirements and respective priorities.

5.2.1 Mobile Application

To guarantee that the mobile application would have acceptable levels of performance, some tests were conducted to its frame rate. Some tests were also conducted to check the robustness of the AR positioning tracking. Before going into more details, some variables should be taken into consideration:

- These tests were always performed on a single device: Xiaomi Mi A3, on Android 11.
- The frame rate heavily depends on how many information has to be rendered to the screen, thus any test made to the frame rate always had the camera pointed in the same direction at the same location.
- The tests were performed on a development build which lacks optimizations.

For the performance tests, it was considered that the mobile application had to maintain in average a value above of fifteen frames per second(fps) in the AR Navigation Scene.

ID	Polygon count range	Average fps
T-1	1000-10000	30
T-2	10000-100000	28
T-3	100000-500000	27
T-4	500000-1000000	18
T-5	>1000000	12

Table 5.1: Results of the performance tests.

As the results suggest, there should be a limit of how many objects we can display in front of the user depending of the polygon count of the 3D models. This should be taken into consideration in the future for the development of a content management application/interface.

As for the robustness tests, it was considered that after moving through two specific paths in the building, there shouldn't be a difference superior to one meter between the real location and the virtual location. These tests were important to understand if the mobile application could rely on the device for its positioning tracking despite suffering from occasional drifting (due to low light conditions), and therefore discard the possibility of having to change the requirement FR-26 and rely on beacons to track the device's location within the building. Most importantly, part of the data structure relied on the results of these tests as well, since the 3D model for each building was potentially going to be partitioned into different floors. For these tests two paths were considered:

- **Path 1**: Crossing a shortest distance but going through a staircase between floors in order to test the device's capabilities in tracking its position vertically. The path is represented in figure 5.9.
- **Path 2**: Crossing a longer distance going through different light conditions in order to test the device's capabilities in tracking its position under different light conditions horizontally. The path is represented in figure 5.10.

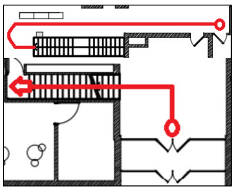


Figure 5.9: Testing path 1.

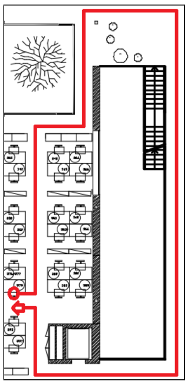


Figure 5.10: Testing path 2.

The results for both paths are shown in table 5.2 and 5.3. All tests surpassed the expectations and got results bellow one meter. This confirmed that for this project there was no need to change any requirement and that a single 3D model of the building was enough for it to work as expected.

ID	Path	Distance(m)	Between Floors	Difference(m)
T-6	1	30	Yes	0,41
T-7	1	30	Yes	0,33
T-8	1	30	Yes	0,36
T-9	1	30	Yes	0,42
T-10	1	30	Yes	0,33
T-11	1	30	Yes	0,37
Difference's Average				0,37
Difference's Standard Deviation				0,0385

Table 5.2: Results of the robustness tests for the first path.

ID	Path	Distance(m)	Between Floors	Difference(m)
T-12	2	62	No	0,84
T-13	2	62	No	0,65
T-14	2	62	No	0,63
T-15	2	62	No	0,54
T-16	2	62	No	0,75
T-17	2	62	No	0,70
Difference's Average				0,685
Difference's Standard Deviation				0,104

Table 5.3: Results of the robustness tests for the second path.

From these results, we can assume that the mobile application fulfills the robustness requirement. The results' averages are below the acceptance value and the standard deviation indicates that most of the tries would be under one meter.

5.2.2 Back-end

Most of the tests done to the Back-end consisted of unit tests in order to validate inputs or file extensions. To assure that the Back-end performance was acceptable, the processing time of the AssetBundles was evaluated for different 3D models, including the 3D model of the building. The considered value of acceptance was 10 minutes since this is an asynchronous process. The results displayed in table 5.4 show that the processing time was always below the acceptance value.

ID	Polygon count	Vertices number	Size(MB)	Processing time(mm:ss)
T-18	511	489	0.026	2:10
T-19	1.273	1.186	0.179	2:13
T-20	14.632	7.639	0.884	2:22
T-21	131.761	66.288	7.5	3:31
T-22	1.467.871	734.173	45.7	5:28

Table 5.4: Results of the Back-end performance tests.

Chapter 6

Conclusion

Critical Software has a system that allows its employees to enjoy AR experiences. Typically, the application is used for events, such as welcoming newcomers or celebrating holidays. However, part of welcoming new employees involves showing them around Critical's offices and getting them to know more about the company's activities and work environment. Even older employees have some difficulties finding meeting rooms because they have to travel between different offices and buildings.

This internship focused on trying to solve this problem with the help of AR technology, implementing a solution that was integrated with the existing AR system. This system, includes a mobile application that allows the user to find the shortest path to a point of interest within a building while displays signage and can be used to show exhibitions of digital objects. Its content can be added asynchronously in a server which grants the possibility to change the user's experience without requiring a new build of the app.

The project started by discussing the methodology and planning the tasks throughout the year. This was followed by a study of the state of the art of augmented reality, which included a study of the technology history and its areas of development. Next, a comparative analysis was made of the technologies available for the development of this project to select the most suitable including a brief analysis of other applications.

After this study, a requirements survey was done, which helped in the decision of the technologies to be used in the implementation of the new functionalities. AR Foundation was the development kit that was chosen for having free access, good documentation, supporting iOS and Android, and for being the tool used in the existing system.

Following this stage, a prototype was implemented and tested in a small environment. To further validate the AR capabilities of this prototype and conduct more tests, it was necessary to model a 3D model that could mimic accurately a Critical Software's building within Unity virtual world. Having the model ready, some tests were performed to its location tracking capabilities with positive results. Such results motivated changes in requirements and therefore in the plans for the semester.

From this stage to the end of this internship, the prototype functionalities were then integrated into the existing system and the required back-end changes were done. Some tests were also performed to validate the system's behaviour.

Furthermore, the system is implemented in such a way that it is possible to make changes and take advantage of future opportunities with the AR technology. Future work could be done on a content management mobile application or on improving the current system capabilities considering that there are still much more to explore since hardware keeps improving every year. Functionalities such as occlusion or other positional tracking techniques could be implemented to complement the current system.

In conclusion, most of the goals of this project were achieved. Despite bringing some technical challenges, they were overcome with the Critical's team support contributing to the author's personal and professional growth. It was a rewarding experience as its development resorted to an emerging technology such as AR.

References

[1] Scott W. Ambler. *The Object Primer: Agile Model-Driven Development with UML 2.0.* Cambridge University Press, Cambridge, UK, 3 edition, 2004.

[2] Jakl Andreas. Arcore image. https://www.andreasjakl.com/. Accessed: 2022-01-06.

[3] Chetan Garg Anuroop Katiyar, Karan Kalra. Marker based augmented reality. *IEICE Transactions on Information Systems*, (2):441–445, December 2015.

[4] APDC. Apdc - realidade aumentada e realidade virtual. https://www.apdc.pt/apdc/seccoes/realidade_aumentada__realidade_virtual. Accessed: 2022-01-06.

[5] Apple. Arkit object detection. https://developer.apple.com/documentation/arkit/content_anchors/scanning_and_detecting_3d_objects. Accessed: 2022-01-06.

[6] ARCore. Arcore documentation. https://developers.google.com/ar/develop. Accessed: 2022-01-06.

[7] ARCore. Arcore fundamentals. https://developers.google.com/ar/develop/fundamentals. Accessed: 2022-01-06.

[8] ARCore. Webxr. https://developers.google.com/ar/develop/webxr. Accessed: 2022-01-06.

[9] ARKit. Arkit documentation. https://developer.apple.com/augmented-reality/arkit/. Accessed: 2022-01-06.

[10] ARKit. Arkit fundamentals. https://arvrjourney.com/basic-arkit-concepts-95e845fbb0a1. Accessed: 2022-01-06.

[11] Artoolkit. Artoolkit. http://artoolkit.sourceforge.net/. Accessed: 2022-01-06.

[12] Aryel. Marker-based ar and markerless ar. https://aryel.io/blog/. Accessed: 2022-01-06.

[13] Atlassian. Agile. https://www.atlassian.com/agile. Accessed: 2022-01-06.

[14] Atlassian. Bitbucket. https://www.atlassian.com/software/bitbucket. Accessed: 2022-01-06.

[15] Atlassian. Confluence. https://www.atlassian.com/software/confluence. Accessed: 2022-01-06.

[16] Atlassian. Jira. https://www.atlassian.com/software/jira. Accessed: 2022-01-06.

[17] Atlassian. Scrum vs kanban. https://www.atlassian.com/de/agile/kanban/kanban-vs-scrum. Accessed: 2022-01-06.

[18] Atlassian. What is Kanban? https://www.youtube.com/playlist?list=PLaD4FvsFdarR3oF1gp5_Nmn1L-BQIE9sW. Accessed: 2022-01-06.

[19] Atlassian. What is Scrum? https://www.youtube.com/playlist?list=PLaD4FvsFdarT0B2yi9byhKWYX1YmrkrpC. Accessed: 2022-01-06.

[20] Autodesk. Autodesk. https://www.autodesk.com/viewers. Accessed: 2022-01-06.

[21] Dom Barnard. Virtual reality history. https://virtualspeech.com/blog/history-of-vr. Accessed: 2022-01-06.

[22] BI0MAJOR and ONIATSERJ. League of legends ar. https://nexus.leagueoflegends.com/en-gb/2017/12/dev-summoning-the-worlds-dragon/. Accessed: 2022-01-06.

[23] Blender. Blender. https://www.blender.org/. Accessed: 2022-01-06.

[24] Cory Bohon. Arkit. https://www.techrepublic.com/article/apples-arkit-everything-the-pros-need-to-know/. Accessed: 2022-01-06.

[25] Kevin Brennan. *A guide to the business analysis body of knowledge®: BABOK® Guide*. International Institute of business analysis, 2009.

[26] Avila-Garzon Cecilia, Bacca-Acosta Jorge, Kinshuk, Duarte Joan, and Betancourt Juan. Augmented reality in education: An overview of twenty-five years of research. 13:3, 2021.

[27] Coates Charlotte. Ar in museums. https://www.museumnext.com/article/how-museums-are-using-augmented-reality/. Accessed: 2022-01-06.

[28] Min Chen. Using ar, the french national museum of natural history is bringing back the dodo. https://jingculturecommerce.com/french-national-museum-of-natural-history-saola-studio-revivre/. Accessed: 2022-01-06.

[29] Alistair Cockburn and Norges Bank. Structuring use cases with goals. 12 1997.

[30] Codebude. Qrcoder 1.4.3. https://www.nuget.org/packages/QRCoder/. Accessed: 2022-01-06.

[31] Julie Deane. Cambridge satchel company. https://www.shopify.com/plus/customers/cambridge-satchel-company. Accessed: 2022-01-06.

[32] Thomas Didier. Ar in football. https://atomicdigital.design/blog/1998-augmented-reality-football. Accessed: 2022-01-06.

[33] Docker. Docker. https://www.docker.com/. Accessed: 2022-01-06.

[34] Bryon Federick. Myron krueger. http://thedigitalage.pbworks.com/w/page/22039083/Myron%20Krueger. Accessed: 2022-01-06.

[35] Al Filreis. Electrocular. http://nineteen-sixty.blogspot.com/2017/03/its-electrocular.html. Accessed: 2022-01-06.

[36] Forbusiness. Business snapchat survey. https://forbusiness.snapchat.com/blog. Accessed: 2022-01-06.

[37] Randall Fulton and Roy Vandermolen. *Airborne Electronic Hardware Design Assurance: A practitioner's guide to RTCA/DO-254*. CRC Press, 2017.

[38] Paul Milgram Fumio Kishino. A taxonomy of mixed virtual visual displays. *IEICE Transactions on Information Systems*, (12):15, December 1994.

[39] The Medical Futurist. Ar in healthcare. https://medicalfuturist.com/augmented-reality-in-healthcare-will-be-revolutionary/. Accessed: 2022-01-06.

[40] Google. Cloud anchors documentation. https://codelabs.developers.google.com/codelabs/arcore-cloud-anchors#0. Accessed: 2022-01-06.

[41] Google. Google ar shopping interest survey. https://www.thinkwithgoogle.com/consumer-insights/consumer-trends/ar-shopping-interest-statistics/. Accessed: 2022-01-06.

[42] Google. Googleglass. https://www.google.com/glass/start/. Accessed: 2022-01-06.

[43] Anders Henrysson, Mark Billinghurst, and Mark Ollila. Ar tennis. *http://doi.acm.org/10.1145/1179849.1179865*, 01 2006.

[44] Hopkinsmedicine. Ar surgery. https://www.hopkinsmedicine.org/news/articles. Accessed: 2022-01-06.

[45] Ikea. Ikea mobile app. https://www.ikea.com/pt/pt/customer-service/mobile-apps/. Accessed: 2022-01-06.

[46] Imgawards. Artennis. https://www.imgawards.com/games/ar-tennis/. Accessed: 2022-01-06.

[47] Informit. Brief history of ar. https://www.informit.com/articles/article.aspx?p=2516729&seqNum=2. Accessed: 2022-01-06.

[48] Michael Jahn. Zxing.net. https://github.com/micjahn/ZXing.Net. Accessed: 2022-08-13.

[49] Matt Lake. Ar in football. https://www.nytimes.com/2000/01/27/technology/when-the-game-s-on-the-line-the-line-s-on-the-screen.html. Accessed: 2022-01-06.

[50] John Mankins. Technology readiness level – a white paper. 01 1995.

[51] Medium. Arcore cloud anchor. https://medium.com/@ardeploy. Accessed: 2022-01-06.

[52] Microsoft. Hololens. https://docs.microsoft.com/en-us/hololens/hololens1-hardware. Accessed: 2022-01-06.

[53] Microsoft. Teams. https://www.microsoft.com/pt-pt/microsoft-teams/group-chat-software. Accessed: 2022-01-06.

[54] Microsoft. Visual studio. https://visualstudio.microsoft.com/. Accessed: 2022-01-06.

[55] David Mizell. Boeing's wire bundle assembly project. 01 2001.

[56] Jeremy M. Norman. Louis rosenburg. https://www.historyofinformation.com/detail.php?id=4231. Accessed: 2022-01-06.

[57] Jeremy M. Norman. The Master Key book. https://historyofinformation.com/detail.php?id=4233. Accessed: 2022-01-06.

[58] Taiichi Ohno. *Toyota Production System: Beyond Large-Scale Production*. Productivity, Portland, OR, 1988.

[59] Overleaf. Overleaf. https://www.overleaf.com/. Accessed: 2022-01-06.

[60] Wayne Piekarski and Bruce Thomas. Arquake: The outdoor augmented reality gaming system. *Commun. ACM*, 45:36–38, 01 2002.

[61] David Pierce. Arcore. https://www.wired.com/story/google-joins-the-augmented-reality-party-with-arcore/. Accessed: 2022-01-06.

[62] Pikkart. Pikkart documentation. https://developer.pikkart.com/servizi/Menu/dinamica.aspx?idSezione=617&idArea=619&idCat=626&ID=696&TipoElemento=categoria. Accessed: 2022-01-06.

[63] Pikkart. Pikkart pricing. https://developer.pikkart.com/augmented-reality/sdk/. Accessed: 2022-01-20.

[64] Akhil Pillai. Mixed reality. https://medium.com/@akilpilai/ar-vr-and-mixed-reality-an-intro-79a516e5b835. Accessed: 2022-01-23.

[65] Bridget Poetker. Augmented reality history. https://www.g2.com/articles. Accessed: 2022-01-06.

[66] Stephanie Ray. The risk management process in project management. https://www.projectmanager.com/blog/risk-management-process-steps. Accessed: 2022-01-06.

[67] Gerhard Reitmayr and Tom W. Drummond. Going out: robust model-based tracking for outdoor augmented reality. pages 109–118, 2006.

[68] Rfcafe. Electrocular - third eye for space explorers (early google glass?). https://www.rfcafe.com/references/popular-electronics/third-eye-space-explorers-july-1962-popular-electronics.htm. Accessed: 2022-01-06.

[69] Robomongo. Robot 3t. https://robomongo.org/. Accessed: 2022-01-06.

[70] L.B. Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. pages 76–82, 1993.

[71] Erik Sherman. Googleglass. https://www.cbsnews.com/news/google-glass-1500-to-buy-80-to-make/. Accessed: 2022-01-06.

[72] Smartinsights. Marketing. https://www.smartinsights.com/digital-marketing-platforms/augmented-reality/5-ways-to-use-augmented-reality-in-your-marketing-strategy/. Accessed: 2022-01-08.

[73] Maximilian Speicher, Brian D. Hall, and Michael Nebeling. *What is Mixed Reality?*, page 1–15. Association for Computing Machinery, New York, NY, USA, 2019.

[74] Nick Statt. Pokemon go. https://www.theverge.com/2016/6/10/11898470/pokemon-go-ios-android-niantic-ingress-ar-first-look. Accessed: 2022-01-06.

[75] Feiner Steven, MacIntyre Blair, Hollerer Tobias, and Webster Anthony. Touring machine. https://graphics.cs.columbia.edu/projects/mars/touring.html. Accessed: 2022-01-06.

[76] I. E. Sutherland. A head-mounted three dimensional display. *Fall Joint Computer Conference, Part I*, (68), December 1968.

[77] Tech27 Systems. Arkit fundamentals. https://tech27.com/resources/point-clouds/. Accessed: 2022-01-06.

[78] Hirotaka Takeuchi and Ikujiro Nonaka. The new new product development game. *Harvard Business Review*, 1986.

[79] TeamGantt. Teamgantt. https://www.teamgantt.com/. Accessed: 2022-01-06.

[80] Theverge. Ar in league of legends. https://www.theverge.com. Accessed: 2022-01-06.

[81] Time. Electrocular. http://content.time.com/time/subscriber/article/0,33009,827272,00.html. Accessed: 2022-01-06.

[82] Tinmith. Arquake. http://www.tinmith.net/arquake/. Accessed: 2022-01-06.

[83] Unity. Arfoundation - camera image access. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/cpu-camera-image.html. Accessed: 2022-04-23.

[84] Unity. Arfoundation - documentation. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html. Accessed: 2022-01-06.

[85] Unity. Arfoundation - features. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.1/manual/index.html. Accessed: 2022-01-06.

[86] Unity. Assetbundles. https://docs.unity3d.com/Manual/AssetBundles-Cache.html. Accessed: 2022-01-06.

[87] Unity. Navmeshcomponents. https://github.com/Unity-Technologies/NavMeshComponents. Accessed: 2022-06-17.

[88] Unity. Probuilder. https://unity.com/features/probuilder. Accessed: 2022-06-17.

[89] Unity. Unity - navmesh components. https://docs.unity3d.com/Manual/NavMesh-BuildingComponents.html. Accessed: 2022-07-22.

[90] Unity. Unity editor. https://unity.com/products/unity-platform. Accessed: 2022-01-06.

[91] Unity. Unity scripting api. https://docs.unity3d.com/ScriptReference/AssetPostprocessor.html. Accessed: 2022-07-08.

[92] Carnegie Mellon University. Software architecture. https://www.sei.cmu.edu/our-work/software-architecture/. Accessed: 2021-01-06.

[93] Fahad Usmani. Assumptions and constraints in project management. https://pmstudycircle.com/assumptions-and-constraints-in-project-management/. Accessed: 2021-01-06.

[94] Dani Vishal. Ar in education.

[95] Visual-Paradigm. Visual paradigm. https://online.visual-paradigm.com. Accessed: 2022-01-06.

[96] Vuforia. Matterport with vuforia documentation. https://library.vuforia.com/features/environments/area-targets/scanning-guide-for-area-targets-using-matterport.html. Accessed: 2022-01-06.

[97] Vuforia. Vuforia documentation. https://developer.vuforia.com/downloads/SDK. Accessed: 2022-01-06.

[98] Vuforia. Vuforia fusion. https://library.vuforia.com/articles/Training/vuforia-fusion-article.html. Accessed: 2022-01-06.

[99] Vuforia. Vuforia pricing. https://www.ptc.com/en/products/vuforia/vuforia-engine/pricing. Accessed: 2022-01-06.

[100] Wearcam. Mann steve. http://wearcam.org/pictures.html. Accessed: 2022-01-06.

[101] Wearcam. Philco headsight. http://wearcam.org/ar/. Accessed: 2022-01-06.

[102] Wikipedia. Ar in football. https://en.wikipedia.org/wiki/1st_%26_Ten_(graphics_system)#/media/File:First-down-line.jpg. Accessed: 2022-01-06.

[103] Wikipedia. Googleglass. https://en.wikipedia.org/wiki/Google_Glass#/media/File:Google_Glass_detail.jpg. Accessed: 2022-01-06.

[104] Wikitude. Wikitude documentation. https://www.wikitude.com/products/wikitude-sdk/. Accessed: 2022-01-06.

[105] Wikitude. Wikitude pricing. https://www.wikitude.com/store/. Accessed: 2022-01-06.

[106] WSCP. Winscp. https://winscp.net/eng/index.php. Accessed: 2022-01-06.

[107] XinReality. Arkit fundamentals. ttps://xinreality.com/. Accessed: 2022-01-06.

Appendices

Appendix A

Use Cases

A.1 Mobile Application Use Cases

Name	Content update (FR-25)
Actor	Mobile Application User
Description	The mobile app user, on opening the mobile application, has the option to update the mobile application content.
Pre-conditions	The mobile device has access to the internet.
Basic Flow	1. The mobile application loads the previously saved content. 2. The mobile application requests the Backend for updated content with the last update date. 3. Backend responds sending back the content changed after the sent date. 4. The mobile application checks if there's any new content that should be downloaded. 5. The mobile application gives the option to update content to the user. 6. The mobile application user chooses to update. 7. The mobile application downloads new content. 8. The mobile application deletes outdated content. 9. The mobile application gives the option to continue to the next view.
Post-conditions	The mobile application gives the option to change to the camera screen and saves the new content successfully.
Alternate Course	2a. Mobile application connection with the Backend fails. 2a.1. Mobile application gives the option of trying to update again (jump to 2.) or to continue with the old content (jump to 9.).

Table A.1: Update the Mobile Application's content.

Name	Choose point of interest to navigate through the building (FR-27)
Actor	Mobile Application User
Description	The mobile app user can choose a point of interest to get assistance while navigating towards it.
Pre-conditions	The mobile application content includes the building required information.
Basic Flow	1. User navigates through the mobile application until reaching the navigation state. 2. User chooses one of the available options from the points of interest's list. 3. Mobile Application shows the path to the selected location. 4. User reaches point of interest location. 5. Mobile Application shows 3D model associated with the selected point of interest.
Post-conditions	User gets to the destination and is allowed to exit the navigation screen.
Alternate Course	3a. Mobile application does not recognize current location due to drifting. 3a.1. Mobile Application user points the camera towards a point of reference. 3a.2. Mobile Application recognizes point of reference and corrects its current position (jump to 3.).

Table A.2: Choose point of interest to navigate through the building.

Name	Scan point of reference (FR-26)
Actor	Mobile App User
Description	The mobile app user scans a point of reference to get into the navigation screen or to correct its position.
Pre-conditions	The mobile application content includes the building required information.
Basic Flow	1. Mobile Application user navigates through the mobile application until reaching the camera state. 2. Mobile Application user points the camera towards a point of reference in the building. 3. Mobile Application recognizes point of reference. 4. Mobile Application opens the navigation screen with its current position.
Post-conditions	Mobile Application opens the navigation screen with the device's current position.
Alternate Course	3a. Mobile application does not recognize current location due to drifting. 3a.1. Mobile Application user points the camera towards a point of reference. 3a.2. Mobile Application recognizes point of reference and corrects its current position (jump to 4.).

Table A.3: Scan point of reference.

Name	View exhibitions and signboards based on indoors location (FR-28)
Actor	Mobile App User
Description	The mobile app user can access augmented reality content based on his indoors location.
Pre-conditions	The mobile application content includes the building required information.
Basic Flow	1. User navigates through the mobile application until reaching the navigation screen. 2. Mobile application displays exhibit components (3D models) and signs based on the current device's location.
Post-conditions	Mobile application shows augmented reality content based on the device's location.
Alternate Course	2a. Mobile application does not recognize current location due to drifting. 2a.1. Mobile Application user points the camera towards a point of reference. 2a.2. Mobile Application recognizes point of reference and corrects its current position (jump to 2.).

Table A.4: View content based on indoors location.

Name	View a map of the building with current position (FR-29, FR-30)
Actor	Mobile Application User
Description	The mobile app user can visualize a representation of its current position inside the building.
Pre-conditions	The mobile application content includes the building required information.
Basic Flow	1. The mobile application user navigates through the mobile application until reaching the navigation state. 2. The mobile application displays an expandable map with the device's current location. 3. The mobile application user selects the map or tilts the device to a horizontal position. 4. The mobile application displays the map in full screen. 5. The mobile application user touches the screen or tilts the device to a vertical position. 6. The mobile application displays the navigation screen.
Post-conditions	Mobile app successfully shows augmented reality content based on the mobile location.
Alternate Course	2a. Mobile application does not recognize current location due to drifting. 2a.1. Mobile Application user points the camera towards a point of reference. 2a.2. Mobile Application recognizes point of reference and corrects its current position (jump to 2.).

Table A.5: View a map of the building with the current position.

A.2 Back-end Use Cases

Name	Create a building (FR-1)
Actor	Content Manager
Description	The content manager can create a building providing its 3d model.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and presses the create building button. 2. The content manager selects the building's 3D model. 3. The content manager fills the rest of the form. 4. The content manager presses the confirm button. 5. The web interface shows the new building in the page.
Post-conditions	The system updates successfully, saving the necessary building's information
Alternate Course	4a. The given data in the form is not valid 4a.1 Backend sends a response with the information of the invalid data error. 4a.2 Web Interface displays the error to the content manager.

Table A.6: Create a new building.

Name	Remove a building (FR-3)
Actor	Content Manager
Description	The content manager can delete a building.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the buildings' list page. 2. The content manager selects the building to delete. 3. The content manager presses the delete button. 4. The web interface shows a new window asking for confirmation. 5. The content manager presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, deleting all the building's information.
Alternate Course	5a. The content manager presses the cancel button.

Table A.7: Remove a building.

Name	Modify a building (FR-2)
Actor	Content Manager
Description	The content manager can modify a building's information or its 3D model.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the buildings' list page. 2. The content manager selects the building to modify. 3. The content manager presses the modify button. 4. The web interface shows a new window with a pre-filled form. 5. The content manager changes the information and presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, changing all the building's information.
Alternate Course	5a. The content manager presses the back button.

Table A.8: Modify a building.

Name	Create a point of interest (FR-4)
Actor	Content Manager
Description	The content manager can create a point of interest.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and presses the create point of interest button. 2. The content manager selects the point of interest's 3D model. 3. The content manager fills the rest of the form. 4. The content manager presses the confirm button. 5. The web interface shows the new point of interest in the page.
Post-conditions	The system updates successfully, saving the necessary point of interest's information
Alternate Course	4a. The given data in the form is not valid 4a.1 Backend sends a response with the information of the invalid data error. 4a.2 Web Interface displays the error to the content manager.

Table A.9: Create a new point of interest.

Name	Remove a point of interest (FR-5)
Actor	Content Manager
Description	The content manager can delete a point of interest.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the points of interest' list page. 2. The content manager selects the point of interest to delete. 3. The content manager presses the delete button. 4. The web interface shows a new window asking for confirmation. 5. The content manager presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, deleting all the point of interest's information.
Alternate Course	5a. The content manager presses the cancel button.

Table A.10: Remove a point of interest.

Name	Modify a point of interest (FR-18)
Actor	Content Manager
Description	The content manager can modify a point of interest's information.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the points of interest' list page. 2. The content manager selects the point of interest to modify. 3. The content manager presses the modify button. 4. The web interface shows a new window with a pre-filled form. 5. The content manager changes the information and presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, changing all the point of interest's information.
Alternate Course	5a. The content manager presses the back button.

Table A.11: Modify a point of interest.

Name	Create a point of reference (FR-6)
Actor	Content Manager
Description	The content manager can create a point of reference.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and presses the create point of reference button. 2. The content manager fills the form. 3. The content manager presses the confirm button. 4. The web interface shows the new point of reference in the page.
Post-conditions	The system updates successfully, saving the necessary point of reference's information
Alternate Course	3a. The given data in the form is not valid 3a.1 Backend sends a response with the information of the invalid data error. 3a.2 Web Interface displays the error to the content manager.

Table A.12: Create a new point of reference.

Name	Remove a point of reference (FR-7)
Actor	Content Manager
Description	The content manager can delete a point of reference.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the points of reference' list page. 2. The content manager selects the point of reference to delete. 3. The content manager presses the delete button. 4. The web interface shows a new window asking for confirmation. 5. The content manager presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, deleting all the point of reference's information.
Alternate Course	5a. The content manager presses the cancel button.

Table A.13: Remove a point of reference.

Name	Modify a point of reference (FR-19)
Actor	Content Manager
Description	The content manager can modify a point of reference's information.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the points of reference' list page. 2. The content manager selects the point of reference to modify. 3. The content manager presses the modify button. 4. The web interface shows a new window with a pre-filled form. 5. The content manager changes the information and presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, changing all the point of reference's information.
Alternate Course	5a. The content manager presses the back button.

Table A.14: Modify a point of reference.

Name	Create a signboard (FR-8)
Actor	Content Manager
Description	The content manager can create a signboard.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and presses the create signboard button. 2. The content manager fills the form. 3. The content manager presses the confirm button. 4. The web interface shows the new signboard in the page.
Post-conditions	The system updates successfully, saving the necessary signboard's information
Alternate Course	3a. The given data in the form is not valid 3a.1 Backend sends a response with the information of the invalid data error. 3a.2 Web Interface displays the error to the content manager.

Table A.15: Create a new signboard.

Name	Remove a signboard (FR-9)
Actor	Content Manager
Description	The content manager can delete a signboard.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the signboard's list page. 2. The content manager selects the signboard to delete. 3. The content manager presses the delete button. 4. The web interface shows a new window asking for confirmation. 5. The content manager presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, deleting all the signboard's information.
Alternate Course	5a. The content manager presses the cancel button.

Table A.16: Remove a signboard.

Name	Modify a signboard (FR-20)
Actor	Content Manager
Description	The content manager can modify a signboard's information.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the signboard's list page. 2. The content manager selects the signboard to modify. 3. The content manager presses the modify button. 4. The web interface shows a new window with a pre-filled form. 5. The content manager changes the information and presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, changing all the signboard's information.
Alternate Course	5a. The content manager presses the back button.

Table A.17: Modify a signboard.

Name	Create a sign (FR-10)
Actor	Content Manager
Description	The content manager can create a sign.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and presses the create sign button. 2. The content manager fills the form. 3. The content manager presses the confirm button. 4. The web interface shows the new sign in the page.
Post-conditions	The system updates successfully, saving the necessary sign's information
Alternate Course	3a. The given data in the form is not valid 3a.1 Backend sends a response with the information of the invalid data error. 3a.2 Web Interface displays the error to the content manager.

Table A.18: Create a new sign.

Name	Remove a sign (FR-11)
Actor	Content Manager
Description	The content manager can delete a sign.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the sign's list page. 2. The content manager selects the sign to delete. 3. The content manager presses the delete button. 4. The web interface shows a new window asking for confirmation. 5. The content manager presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, deleting all the sign's information.
Alternate Course	5a. The content manager presses the cancel button.

Table A.19: Remove a sign.

Name	Modify a sign (FR-21)
Actor	Content Manager
Description	The content manager can modify a sign's information.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the sign's list page. 2. The content manager selects the sign to modify. 3. The content manager presses the modify button. 4. The web interface shows a new window with a pre-filled form. 5. The content manager changes the information and presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, changing all the sign's information.
Alternate Course	5a. The content manager presses the back button.

Table A.20: Modify a sign.

Name	Create an exhibition (FR-12)
Actor	Content Manager
Description	The content manager can create an exhibition.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and presses the create exhibition button. 2. The content manager fills the form. 3. The content manager presses the confirm button. 4. The web interface shows the new exhibition in the page.
Post-conditions	The system updates successfully, saving the necessary exhibition's information
Alternate Course	3a. The given data in the form is not valid 3a.1 Backend sends a response with the information of the invalid data error. 3a.2 Web Interface displays the error to the content manager.

Table A.21: Create a new exhibition.

Name	Remove an exhibition (FR-13)
Actor	Content Manager
Description	The content manager can delete an exhibition.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the exhibition's list page. 2. The content manager selects the exhibition to delete. 3. The content manager presses the delete button. 4. The web interface shows a new window asking for confirmation. 5. The content manager presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, deleting all the exhibition's information.
Alternate Course	5a. The content manager presses the cancel button.

Table A.22: Remove an exhibition.

Name	Modify an exhibition (FR-22)
Actor	Content Manager
Description	The content manager can modify an exhibition's information.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the exhibition's list page. 2. The content manager selects the exhibition to modify. 3. The content manager presses the modify button. 4. The web interface shows a new window with a pre-filled form. 5. The content manager changes the information and presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, changing all the exhibition's information.
Alternate Course	5a. The content manager presses the back button.

Table A.23: Modify an exhibition.

Name	Create an exhibit object (FR-14)
Actor	Content Manager
Description	The content manager can create an exhibit object.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and presses the create exhibit object button. 2. The content manager fills the form. 3. The content manager presses the confirm button. 4. The web interface shows the new exhibit object in the page.
Post-conditions	The system updates successfully, saving the necessary exhibit object's information
Alternate Course	3a. The given data in the form is not valid 3a.1 Backend sends a response with the information of the invalid data error. 3a.2 Web Interface displays the error to the content manager.

Table A.24: Create a new exhibit object.

Name	Remove an exhibit object (FR-15)
Actor	Content Manager
Description	The content manager can delete an exhibit object.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the exhibit object's list page. 2. The content manager selects the exhibit object to delete. 3. The content manager presses the delete button. 4. The web interface shows a new window asking for confirmation. 5. The content manager presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, deleting all the exhibit object's information.
Alternate Course	5a. The content manager presses the cancel button.

Table A.25: Remove an exhibit object.

Name	Modify an exhibit object (FR-23)
Actor	Content Manager
Description	The content manager can modify an exhibit object's information.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the exhibit object's list page. 2. The content manager selects the exhibit object to modify. 3. The content manager presses the modify button. 4. The web interface shows a new window with a pre-filled form. 5. The content manager changes the information and presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, changing all the exhibit object's information.
Alternate Course	5a. The content manager presses the back button.

Table A.26: Modify an exhibit object.

Name	Create an exhibit component (FR-16)
Actor	Content Manager
Description	The content manager can create an exhibit component.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and presses the create exhibit component button. 2. The content manager fills the form. 3. The content manager presses the confirm button. 4. The web interface shows the new exhibit object in the page.
Post-conditions	The system updates successfully, saving the necessary exhibit component's information
Alternate Course	3a. The given data in the form is not valid 3a.1 Backend sends a response with the information of the invalid data error. 3a.2 Web Interface displays the error to the content manager.

Table A.27: Create a new exhibit component.

Name	Remove an exhibit component (FR-17)
Actor	Content Manager
Description	The content manager can delete an exhibit component.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the exhibit component's list page. 2. The content manager selects the exhibit component to delete. 3. The content manager presses the delete button. 4. The web interface shows a new window asking for confirmation. 5. The content manager presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, deleting all the exhibit component's information.
Alternate Course	5a. The content manager presses the cancel button.

Table A.28: Remove an exhibit component.

Name	Modify an exhibit component (FR-24)
Actor	Content Manager
Description	The content manager can modify an exhibit component's information.
Pre-conditions	The content manager has internet and permission to access the necessary information.
Basic Flow	1. The content manager navigates through a web interface and reaches the exhibit component's list page. 2. The content manager selects the exhibit component to modify. 3. The content manager presses the modify button. 4. The web interface shows a new window with a pre-filled form. 5. The content manager changes the information and presses the confirm button. 6. The web interface shows a success message to the content manager.
Post-conditions	The system updates successfully, changing all the exhibit component's information.
Alternate Course	5a. The content manager presses the back button.

Table A.29: Modify an exhibit component.

Appendix B

Functional Requirements

B.1 Backend Functional Requirements

ID	Name	Description	Priority
FR-1	Create a building	The Content Manager must be able to create a building.	Must Have
FR-2	Modify a building	The Content Manager must be able to modify a building.	Must Have
FR-3	Remove a building	The Content Manager must be able to remove a building.	Must Have
FR-4	Create a point of interest	The Content Manager must be able to create a point of interest.	Must Have
FR-5	Remove a point of interest	The Content Manager must be able to remove a point of interest.	Must Have
FR-6	Create a point of reference	The Content Manager must be able to create a point of reference.	Must Have
FR-7	Remove a point of reference	The Content Manager must be able to remove a point of reference.	Must Have
FR-8	Create a signboard	The Content Manager must be able to create a signboard.	Must Have
FR-9	Remove a signboard	The Content Manager must be able to remove a signboard.	Must Have
FR-10	Create a sign	The Content Manager must be able to create a sign in a signboard.	Must Have
FR-11	Remove a sign	The Content Manager must be able to remove a sign from a signboard.	Must Have

Table B.1: Backend Functional requirements with high priority.

ID	Name	Description	Priority
FR-12	Create an exhibition	The Content Manager must be able to create an exhibition.	Should Have
FR-13	Remove an exhibition	The Content Manager must be able to remove an exhibition.	Should Have
FR-14	Create an exhibit object	The Content Manager must be able to create an exhibit object.	Should Have
FR-15	Remove an exhibit object	The Content Manager must be able to remove an exhibit object.	Should Have
FR-16	Create an exhibit component	The Content Manager must be able to create an exhibit component in an exhibition.	Should Have
FR-17	Remove an exhibit component	The Content Manager must be able to remove an exhibit component in an exhibition.	Should Have

Table B.2: Back-end Functional requirements with medium priority.

ID	Name	Description	Priority
FR-18	Modify a point of interest	The Content Manager must be able to modify a point of interest.	Could Have
FR-19	Modify a point of reference	The Content Manager must be able to modify a point of reference.	Could Have
FR-20	Modify a signboard	The Content Manager must be able to modify a signboard.	Could Have
FR-21	Modify a sign	The Content Manager must be able to modify a sign.	Could Have
FR-22	Modify an exhibition	The Content Manager must be able to modify an exhibition.	Could Have
FR-23	Modify an exhibit object	The Content Manager must be able to modify an exhibit object.	Could Have
FR-24	Modify an exhibit component	The Content Manager must be able to modify an exhibit component.	Could Have

Table B.3: Back-end Functional requirements with low priority.

B.2 Mobile Application Functional Requirements

ID	Name	Description	Priority
FR-25	Update Building's App Content	The Mobile Application must be able to update its building's related content.	Must Have
FR-26	Read QR Codes	The Mobile Application must be able to read and interpret QR Codes.	Must Have
FR-27	Show Shortest Path	The Mobile Application must be able to calculate and show the shortest path to a destination.	Must Have
FR-28	Show Building's Content	The Mobile Application must be able show 3D models in designated locations within the building.	Must Have

Table B.4: Mobile Application Functional requirements with high priority.

ID	Name	Description	Priority
FR-29	Show a Map of the Building	The Mobile Application must be able to show a representation of the building and the mobile user's location.	Should Have

Table B.5: Mobile Application Functional requirements with medium priority.

ID	Name	Description	Priority
FR-30	Screen Transition tilting phone	The Mobile Application must be able to transition between the map and the AR view according to the device's rotation.	Could Have

Table B.6: Mobile Application Functional requirements with low priority.