Collaborative use of virtual and augmented reality in selected domains

Martin Piliar[*]
Supervised by: Ján Lacko[†]

Faculty of Informatics
Pan-european university
Bratislava / Slovakia

Abstract

It is difficult for an external observer to focus on observing the activity of the virtual reality user through the display and at the same time to observe his movement in space. In various applications of virtual reality patient therapy, it is necessary for the therapist to monitor the patient and at the same time his activity in the virtual environment. In our solution, we focus on the use of MS Hololens to expand the virtual space used by the user in virtual reality, so that it is possible to monitor the user and his activity in VR. Our work consists of implementing a virtual and augmented reality network connection that makes it possible for an external observer to break through these technological barriers and see a virtual representation of the user over the real world. Along with the holographic representation of the virtual user, it is also possible to see all selected interactable objects with which the virtual reality user interacts. This linking is done by loading an AR tag that represents the center point of the virtual world.

Keywords: virtual reality, augmented reality, networking, multiuser, cooperation

1 Introduction

The increasing use of virtual reality is opening up new opportunities for various industrial, medical and entertainment sectors. In industrial sectors, virtual reality can simulate various working conditions without endangering human lives at a fraction of the cost. In the medical sector, virtual reality helps, for example, in rehabilitation after severe injuries or even after a stroke [1]. The entertainment industry is one of the main drivers of the commercialisation of virtual reality. This industry benefits from the use of virtual reality mainly due to the availability of games that require a certain level of activity, unlike traditional computer games [2].

However, due to its strong immersion, virtual reality limits external users from interacting with the virtual world, as it is not possible to see the virtual and real world simultaneously when using two virtual reality devices. This fact can be limiting especially for industrial and medical solutions, where it is sometimes necessary to see, for example, human behaviour and reactions in a given virtual environment. One solution to this problem may be, for example, the use of augmented reality in the form of portable devices such as mobile phones or tablets, but these may present shortcomings in the form of camera image delays, poor camera quality, and so on. The use of mixed reality in the form of wearable devices such as Microsoft Hololens or MagicLeap can address this shortcoming.

This paper will focus on the implementation to align the virtual reality world with the real world using mixed reality devices and AR markers. It also provides guidance on the development of a platform used to connect virtual and augmented reality using commercially available solutions, both software and hardware, that have already been developed. The paper also aims to provide insight into the potential application of this platform in practice.

In Section 2, we review the most successfully implemented virtual reality interfaces together with mixed or augmented reality. In particular, we will look at the quality and distinctiveness of each implementation. Our proposed implementation is described in Section 3, in which we describe our method of implementing the networking and optical interconnection of virtual and mixed reality using commercially available technology solutions such as the Vuforia Engine[1] or the Photon Network Engine[2].

2 Related Work

Interconnecting virtual and augmented reality is not an entirely new concept. According to paper by Pidel and Ackermann [3], the different ways of connecting can be divided into several categories:

- **Synchronous and in the same place:** Collaborators interact with each other in one place in real time. An

[*] xpiliar@paneurouni.com
[†] jan.lacko@paneurouni.com

[1] `https://library.vuforia.com/`

[2] `https://doc.photonengine.com/en-us/realtime/current/getting-started/realtime-intro`

example would be teaching in high schools or colleges.

- **Synchronous but remotely:** Collaborators interact with each other in real time but are not in the same location.

- **Asynchronous and remotely:** This form of collaboration includes many common communication methods such as chat or email. Thus, collaborators can work at times and situations that are convenient for them.

- **Asynchronous and in the same place:** A more specific and less common implementation in which real-time interaction is not essential, but in which the location in which both participants are located is essential.

The authors of the paper also elaborated on the various advantages and disadvantages of different forms of interconnection, such as possible information overload in asynchronous and remote forms of interconnection.

The use of a synchronised on-site collaborative system [3] was also used in some form by [4], in which they describe the interconnection of a virtual reality device with augmented reality on a mobile device. The synchronization of tracked objects such as hands, head or objects that the virtual reality user interacts with is done by using the Photon Unity Networking platform. Alignment of the virtual object in augmented reality is done by reading the AR tag through the mobile device's camera, which is then processed using natural feature tracking (NFT) technology [5].

This method consists of detecting features such as edges, borders, etc. in a given image. Such features are found naturally in every image [6]. This method makes it possible to use different sufficiently distinct images without the need to design specific identification tags.

Also, Frantz et al [7]. studied the effect of combining automatic marker position calibration using the Vuforia Engine together with Spatial mapping of the Microsoft Hololens device. Their results show that automatic alignment of the object using the Vuforia Engine achieved an average perceived holographic shift of 1.41mm, 68% less than manual registration, which utilized only the Spatial mapping of the Hololens device.

In their research, Sramka et al [1]. studied combining traditional rehabilitation methods along with the use of virtual reality to rehabilitate patients after stroke. In their research they focused on rehabilitation of the most common disorders caused by stroke such as sensory disturbances-disorders of symbolic functions- cognitive impairment. For each category of disorders, a specific environment was created to target a particular symptom of the disorder.

3 Proposed method

Due to the ease of scalability and high availability of quality frameworks required for developing collaborative applications for virtual and mixed reality, we present our proposed method by leveraging common commercially available devices and tools in the market.

Our proposed implementation consists of two separate projects created in the Unity3D game engine version 2020.3.20f. A slightly outdated version of Unity3D is used due to compatibility issues with the Mixed Reality Toolkit in newer Long Term Support (LTS) versions of Unity3D. By implementing the virtual and mixed reality applications separately, it is possible to avoid potential problems associated with incompatibility between the SteamVR Software Development Kit (SDK)[3] and the Microsoft Mixed Reality Toolkit SDK[4].

3.1 Real-time synchronization

Based on the findings from [4], we observe that the use of the Photon Unity Networking solution allows us to easily manage interactive objects as well as to synchronize rigid-body objects in a high-quality and reliable manner. Connecting the two devices relies on the use of the same application ID, thanks to which the Photon Engine NameServer is able to automatically initiate a connection between two distinct applications. The Photon Engine Nameserver and LoadBalancer can be self-hosted or one of Photon Engine's server solutions can be used, allowing us to easily migrate the server as needed and available.

Both applications use the User Datagram Protocol (UDP) for connectivity, which ensures lower latency between the two devices [8] as the wireless connection is used when using the Microsoft Hololens 2 mixed reality headset. By using the UDP transmission protocol, we reduce latency and mitigate any potential jitter caused by packet loss by smoothing motion along with predicting the motion and rotation of the object.

The connection of both devices via the Photon Network is as follows:

1. Both devices connect to the Photon Master Server in an unspecified order.

   Note: The connection method and the endpoint of the Photon Master Server is determined by the configuration file settings provided by the Photon Engine.

2. The application running on the virtual reality device always behaves as a client and searches for an available room at a certain time interval.

[3] https://valvesoftware.github.io/steamvr_unity_plugin/

[4] https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/

3. The mixed reality device always behaves as a master client that creates the room.
4. Once the room is created by the master client, the client connects to it as soon as it is possible.
5. Once the connection is successfully established, the master client sends the relevant data to the client. The method of sending and the content of the data is described in more detail in subsection 3.4 .

3.2 Mixed reality alignment

Determining the center point of the virtual world in the real world is one of the critical points of interconnecting virtual and mixed reality. The centre point for a virtual reality user is determined by a defined ’play area’, which represents a safe zone for a user wearing a headset on their head. Depending on the headset used, the play area has different characteristics and requirements, for example, in the case of a room-scale setup of the HTC VIVE headset, the minimum required play area is 2m x 1.5m.[5]

In our proposed implementation, the center point is determined by the corresponding AR marker, thanks to which the rotation, position and height of the virtual world in the real one can be determined. The registration of the marker is done by the augmented reality tool Vuforia Engine, which registers each marker using natural feature tracking technology. Thanks to this technology, it is possible to use conventional images as markers or even generated marker IDs. PTC Inc., the developer of the Vuforia Engine, provides pre-processing of the markers and subsequent ”rating” of one to five stars based on their proprietary methods [9]. This rating lets us know in advance whether our applied marker meets the values required by the Vuforia Engine. The basic criteria for evaluating a marker are its resolution along with the number of features generated.

As can be seen from the first glance in Figure 1, the 5-star marker consists of a relatively large number of edges and corners, and at the same time has the potential for a strong contrast against the surroundings on which it will be placed. In the other hand, for the 1-star marker shown in Figure 1, we can see that the Vuforia Engine algorithm has only recognized a few features, the number of which is not sufficient for high quality and reliable tracking of the marker.

The application running on the Microsoft Hololens 2 does not need to continuously track the marker using the Vuforia Engine, as the Microsoft Hololens device features Spatial mapping, which continuously constructs a virtual geometry of the real world, making the device aware of its relative position and rotation in space[10]. The Vuforia Engine tool works in conjunction with the Microsoft Hololens’ spatial tracking feature. The use of this cooperation provides automatic and seamless switching between a mode whereby the position of the marker is determined in the combined mode of the Vuforia Engine and the spatial mapping function or using only the spatial tracking function of the Microsoft Hololens device.

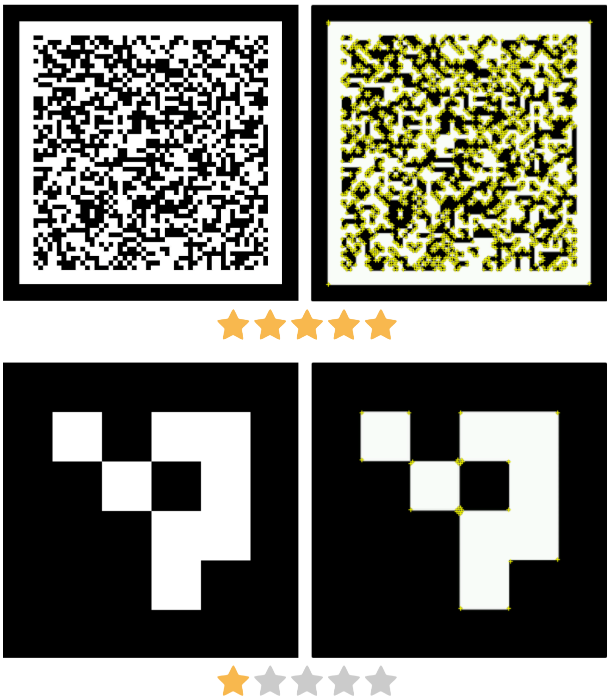

Figure 1: An example of a ”5 star” (top)[6] and ”1 star” (bottom)[7] generated AR tags.

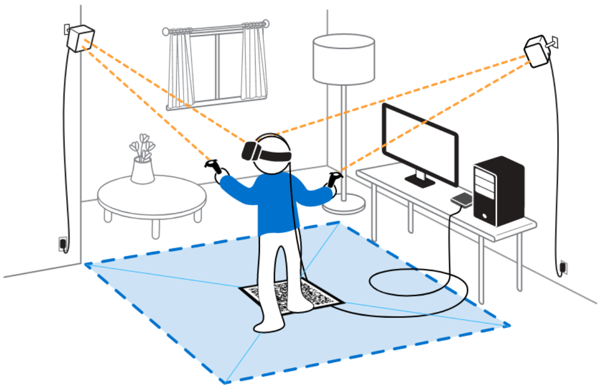

Figure 2: Placement of a marker in the centre of the Play Area. *(Image courtesy of HTC Corp.)*

As can be seen in Figure 2, the selected AR marker needs to be placed in the center of the safe play zone, whose size, shape and position is determined by the virtual reality device currently in use. In the case of using the HTC VIVE, the safe play zone can be set during the initial setup of the device, during which critical points of the

[5] https://www.vive.com/uk/support/vive/category_howto/planning-your-play-area.html

[6] https://shawnlehner.github.io/ARMaker/

[7] https://chev.me/arucogen/

zone are determined using the controllers. This makes it easy to mirror, for example, a pre-prepared and pre-drawn zone in the real world.

The rotation of the AR marker represents the forward Z-axis, which the virtual reality user determines along with defining the game zone during the initial setup of the device. The AR marker is not required to be placed at the center point during the entire run of the application, but in the current implementation, to remove it, it is required that the cameras of the Microsoft Hololens device cannot see the marker during the removal process. If the AR marker is removed, the application will start to use only the spatial mapping of the Microsoft Hololens device during which holograms may drift as mentioned in paper [7].

3.3 User experience

During a more comprehensive interaction with Microsoft Hololens 2, we found some limitations that could hinder users when working with our application. One such limitation is working in 3D space, where the user is sometimes forced to turn around for various modal dialogues or other situations happening in the application. Another potential obstacle we encountered in our work with Microsoft Hololens is the rather slow typing on the integrated keyboard. Marcus Ghaly, a senior holographic designer at Microsoft, addresses some of the issues we mentioned in his case study[8], but since the Hololens platform and also common mixed reality interaction is still in its infancy, we decided to take advantage of traditionally used methods of user experience implementation.

As mentioned in subsection 3.1, the Microsoft Hololens device takes the role of a master client, which allows it to create and close network rooms. Since we have chosen to demonstrate our implementation of the linking of virtual and mixed reality in the form of a platform used in physiotherapy sessions, it is necessary to facilitate the user experience for the user wearing the mixed reality device. This step is implemented through a combination of leveraging the predefined UI elements and UX guidelines provided in MRTK and implementing a cross-platform application in which a user working with Microsoft Hololens can perform more complex operations. The use of predefined UI elements allows us to unify the visual appearance of the application along with the operating system of the Microsoft Hololens 2 device, creating a seamless transition for the user from the operating system to our application.

3.4 User data management

The subject of the need to use an external companion application in order to simplify the user experience during the use of the Microsoft Hololens device has already been addressed in subsection 3.1. In this subsection we describe the management and the way of transferring various data from the mentioned companion application towards the Microsoft Hololens device.

All user data is stored on the Google Firebase Realtime Database service, which provides a NoSQL based backend-as-a-service [11]. The data is stored in a structured JavaScript Object Notation (JSON) object consisting of key/value pairs [12]. Values are used for primitive data types (integer, string, float, etc.), but also for complex objects. One of the main reasons we decided to use a real-time NoSQL database instead of a classic relational SQL database is the fast update of data across devices. It is also easier for us to manage data across different implementations and platforms. The ease of parsing the JSON data we get from the database is another reason to use a NoSQL database since the Windows platform is not supported by Google Firebase Unity3D SDK[9]. We use Representational state transfer (REST) API to connect to the server and manipulate the data [13].

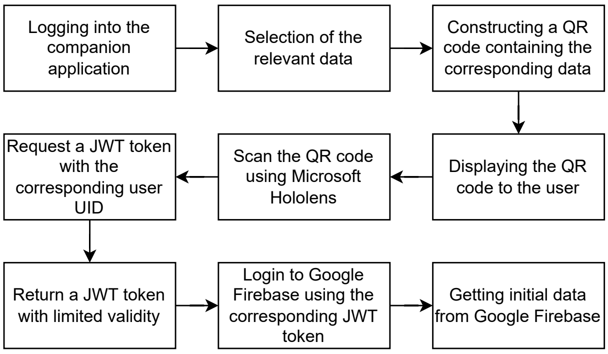
Logging into the companion application
Selection of the relevant data
Constructing a QR code containing the corresponding data
Displaying the QR code to the user
Scan the QR code using Microsoft Hololens
Request a JWT token with the corresponding user UID
Return a JWT token with limited validity
Login to Google Firebase using the corresponding JWT token
Getting initial data from Google Firebase

Figure 3: The process of logging in with Microsoft Hololens and getting data from Google Firebase.

Figure 3 shows the process by which the Microsoft Hololens device connects to the Google Firebase back-end. The whole process starts by using a companion application that the user logs into using pre-created credentials. In the companion application, after selecting the relevant data, a QR code containing the data along with a unique user ID is generated.

In the demonstrated version of the app for Microsoft Hololens, user login is required and the app actively searches for the QR code from startup using the built-in cameras on the device. If the QR code is successfully read by the device, the data string is parsed, and a request for a custom authentication back-end is created, which functionality is described in more detail in subsection 3.4.1. If the data sent to the back-end is valid, the back-end will return a valid JSON Web Token (JWT) token to the device, which can be used to log into the Google Firebase database, and which will also be used to authenticate the user in any data transfers [14].

In Figure 4 we can see that the data structure required

[8] https://bit.ly/ms-case-study

[9] https://firebase.google.com/docs/unity/setup

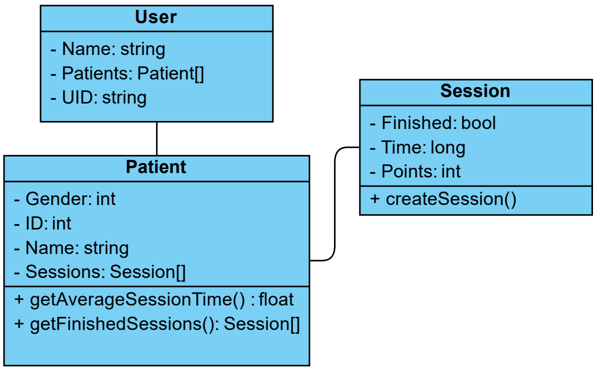

Figure 4: Class diagram for demonstrated data.

by the demonstrated application is quite simple. The illustrated data structure is mainly used to transfer basic information about the user and their sessions between all devices used. The session data is mainly informative, but the application user data has exclusively practical uses such as authentication to the database or displaying all their patients. The user class contains the name of the user of the mixed reality device, his patient list and his UID, which is resent along with the authentication token to the custom back-end in case of JWT token expiration for authentication renewal. The user's unique ID also serves as an access identifier that defines data access and thus ensures that a given user has access only to their respective data. The patient class contains basic information about virtual reality users falling under the mixed reality user. To find out basic information about interactions between users, the patient class contains basic data about individual sessions. The patient and session classes are not required to use the standalone platform and are for demonstration purposes of this application only.

3.4.1 Custom authentication back-end

For the reasons described in subsection 3.3, it is necessary to ensure secure and active data transfer between the companion application and the Microsoft Hololens device. The use of a QR code alone, which would contain all the necessary data, would close the possibility of two-way data transfer and would also pose a potential problem in the case of a larger data transfer, since the maximum capacity of a QR code is approximately 3kb. Despite the fact that the Google Firebase Admin SDK supports the C# programming language, which is used by Unity3D, we decided to implement the custom authentication back-end separately in the Python programming language using the Flask micro-web framework [15].

By implementing a separate authentication back-end, we can easily manage and control individual authentication requests in case multiple devices authenticate at the same time. The user authentication back-end includes, among other things, an endpoint that takes the UID of the user requesting authentication as its query parameter. After successful processing and authentication of the data, a JWT token is returned to the user, which is valid for one hour.

4 Examples

The interconnection of the different tools, solutions and applications we have described so far is demonstrated on a platform with a potential use for physiotherapists and their patients. However, since the goal of the work is not to create a final application for the purpose of real deployment, a lot of the shown functionalities and features have been implemented only for the purpose of demonstrating one of the many potential uses of the platform.

The core components of the platform are the interconnection of virtual and mixed reality devices using the Photon Network solution, AR tag loading using the Vuforia Engine service, and simple data management using a custom authentication backend and Google Firebase database service.

4.1 Virtual and mixed reality applications

The virtual reality application consists of two rooms that the user can move around and interact with. The lobby room is mainly used to familiarize the user with the device. While waiting for a connection, the user can move around the room and interact with some of the objects at will. As shown in Figure 5, once the connection to the mixed reality is established, the user is shown on a virtual screen his previous statistics from any previous sessions, which they can view until the physiotherapist starts the session.

The second room demonstrates in an entertaining way the results of our virtual and mixed reality integration.In a mini-game called ”PunchIt”, the user is shown at 1-5 second intervals circles that are rotated on him and into which he has to punch with his hands, which are represented by boxing gloves. The main objective of the mini-game is to make the user hit the displayed circles as hard as possible to score the most points in a given time period. The game demonstrates the alignment of virtual hands with real ones in a creative way, while also demonstrating simple synchronization of the user's interactions with the in-game objects.

As can be seen from Figure 7, the user's interaction with the mini-game can be viewed through the Microsoft Hololens device, in which virtual gloves can be seen deployed and aligned on the virtual reality user's hands, along with circles spawned in the same position on both devices. The user wearing the Microsoft Hololens device can move at any safe distance around the virtual reality user during observation.

Our goal was to maintain as much visual consistency as possible between the devices, so despite the lower performance of the Microsoft Hololens device, both users are

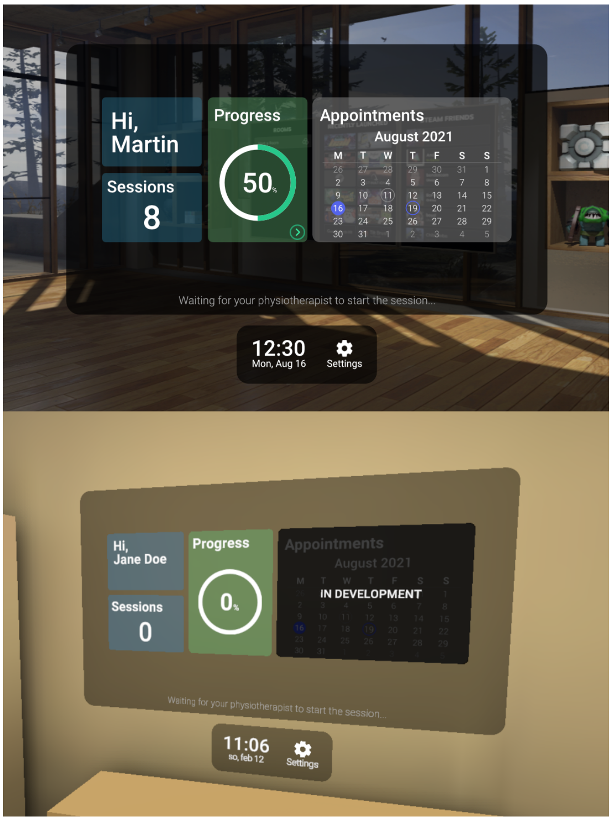

Figure 5: Design and the implementation of a dashboard with which the virtual reality user can interact.

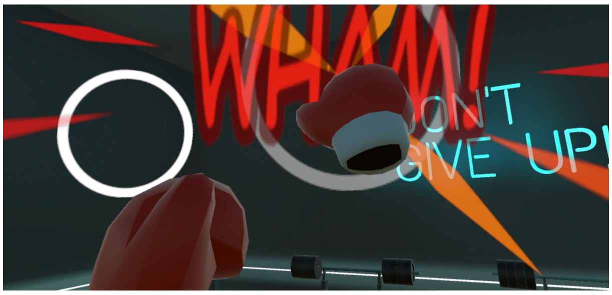

Figure 6: Screenshot from playing the ”PunchIt” mini-game.

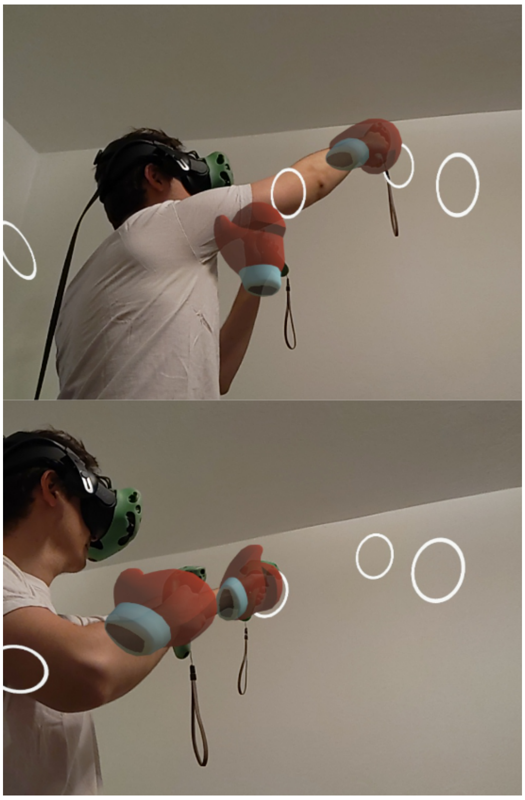

Figure 7: The view from the Microsoft Hololens device. Due to the different heights between the eyes and cameras of the device, a slight offset is created in the screenshots, the user sees the models aligned correctly.

shown identical 3D models and animations as can be seen in Figure 7 and Figure 6.

4.2 Companion application

We described the functionality and reasons for using the companion application in sections 3.4 and 3.3. The content and design of the companion application was designed for a particular platform demonstration. The process of working with the application can be seen in Figure 8. The whole process starts with the user logging in using the precreated data that has been created in the Google Firebase platform. The application itself does not have an option to create a user account. This step was implemented to expedite the development of the demonstrated application without compromising its security. Upon successful login, the user is presented with a customizable dashboard, displaying various statistics and data about the patients or their sessions. The most important part of the application is a tab called ”Patients”, the contents of which are all active patients of the currently logged in user.

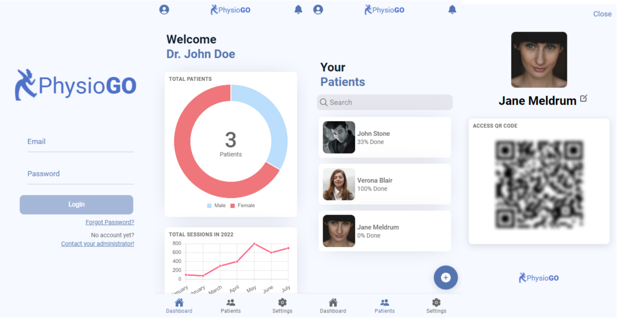

Figure 8: Preview of the workflow with the companion application.

Patients’ names are fictional for demonstration purposes and their photos are used from freely available sources[10]. By selecting the relevant patient, the user is presented with a modal window which again summarizes the basic patient data (name, photo) along with a generated QR access code containing the data described in section 3.4.

5 Conclusion

In this work, we have described the implementation of virtual and mixed reality interconnection using commercially available frameworks and devices. By implementing tools such as Photon Engine and Vuforia Engine, we have made the platform more flexible and opened it up to potentially faster and better scalability. We then demonstrated our developed implementation on a potential platform whose applications can be found, for example, in physiotherapy or other medical or industrial sectors.

Despite the fact that we have been successful in using commercially available technologies, it would be appropriate for the implemented platform to be fully accessible without Internet access. In the future, we would like to eliminate this problem by implementing some solutions ourselves that do not require an active internet connection. One potential solution could be to combine the database backend with the authentication backend and replace the Google Firebase database with a custom solution, or by implementing another standalone hosted solution.

References

[1] Miron Sramka, Ján Lacko, Eugen Ruzicky, and Ján Mašán. Combined methods of rehabilitation of patients after stroke: Virtual reality and traditional approach. *Neuro endocrinology letters*, 41:123–133, 07 2020.

[2] Xiao Yu. Research and practice on application of virtual reality technology in virtual estate exhibition. *Procedia Engineering*, 15:1245–1250, 12 2011.

[3] Catlin Pidel and Philipp Ackermann. *Collaboration in Virtual and Augmented Reality: A Systematic Overview*, pages 141–156. 08 2020.

[4] Muhammad Nor’a and Ajune Ismail. Integrating virtual reality and augmented reality in a collaborative user interface. *International Journal of Innovative Computing*, 9, 11 2019.

[5] U. Neumann and S. You. Natural feature tracking for augmented reality. *IEEE Transactions on Multimedia*, 1(1):53–64, 1999.

[6] Saša Ćuković, Michele Gattullo, Frieder Pankratz, Goran Devedzic, Ernesto Carrabba, and Khelifa Baizid. Marker based vs. natural feature tracking augmented reality visualization of the 3d foot phantom. 01 2015.

[7] Taylor Frantz, Bart Jansen, Johnny Duerinck, and Jef Vandemeulebroucke. Augmenting microsoft’s hololens with vuforia tracking for neuronavigation. *Healthcare Technology Letters*, 5, 09 2018.

[8] Fahad Al-Dhief, Naseer Sabri, and Musatafa Albadr. Performance comparison between tcp and udp protocols in different simulation scenarios. *International Journal of Engineering & Technology*, 12 2018.

[9] Xinqi Liu, Young-Ho Sohn, and Dong-Won Park. Application development with augmented reality technique using unity 3d and vuforia. 2018.

[10] Patrick Hübner, Kate Clintworth, Qingyi Liu, Martin Weinmann, and Sven Wursthorn. Evaluation of hololens tracking and depth sensing for indoor mapping applications. *Sensors*, 20:1021:1–23, 02 2020.

[10] CC0 Avatar placeholders `https://pravatar.cc/`

[11] Margaretha Ohyver, Jurike V. Moniaga, Iwa Sungkawa, Bonifasius Edwin Subagyo, and Ian Argus Chandra. The comparison firebase realtime database and mysql database performance using wilcoxon signed-rank test. *Procedia Computer Science*, 157:396–405, 2019. The 4th International Conference on Computer Science and Computational Intelligence (ICCSCI 2019) : Enabling Collaboration to Escalate Impact of Research Results for Society.

[12] Nurzhan Nurseitov, Michael Paulson, Randall Reynolds, and Clemente Izurieta. Comparison of json and xml data interchange formats: A case study. pages 157–162, 01 2009.

[13] Li Li and Wu Chou. Design and describe rest api without violating rest: A petri net based approach. In *2011 IEEE International Conference on Web Services*, pages 508–515, 2011.

[14] Michael Jones, Brian Campbell, and Chuck Mortimore. JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants. RFC 7523, May 2015.

[15] Nurul Shazwin Idris, Cik Feresa Mohd Foozy, and Palaniappan Shamala. A generic review of web technology: Django and flask. 2019.