Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Master in Human Computer Interaction and Design

Master Thesis

Ampliación de los sistemas de lenguaje de diseño 2D para XR: un enfoque centrado en el usuario para crear y adaptar componentes de interacción para entornos inmersivos

Extending 2D Design Language Systems for XR: A User-Centered Approach to Creating and Adapting Interaction Components for Immersive Environments

Author: Yağmur Eryılmaz
Madrid, Julio, 2025 /July, 2025

Supervisor:

Angélica De Antonio Jimenez

Director of the Research Group on Human Computer Interaction and Advanced Interactive Systems

Universidad Politécnica de Madrid

Co-Supervisor:

Robby Van Delden

Assistant Professor

University of Twente

Company Supervisor:

Jean-Marc Huijskens

Lead Architect AR&VR

Philips, Netherlands

Resumen

Las tecnologías de Realidad Extendida (XR) introducen interacciones espaciales inmersivas que desafían los marcos de diseño 2D tradicionales utilizados en el desarrollo de productos digitales. En Philips, el Sistema de Lenguaje de Diseño (DLS) garantiza la coherencia y la facilidad de uso en todas las plataformas, pero requiere una adaptación para soportar eficazmente los entornos XR. Esta tesis emplea un enfoque centrado en el usuario para hacer evolucionar el DLS de Philips identificando los componentes XR que faltan y traduciendo los elementos 2D existentes en experiencias inmersivas 3D. La investigación descubre las limitaciones actuales y las necesidades de los usuarios a través de la revisión de la literatura, análisis de mercado, entrevistas y pruebas de usuario. Los métodos participativos, incluidos los talleres de codiseño y la clasificación de tarjetas, contribuyen a la ideación de nuevos componentes de interacción. Se crearon prototipos de componentes de interacción gestual, retroalimentación basada en la mirada y navegación espacial en 3D. La directriz resultante tiene como objetivo proporcionar patrones de diseño XR escalables y reutilizables alineados con la visión estratégica de Philips, fomentando la mejora de la usabilidad y la coherencia en las aplicaciones XR.

Palabras clave:

Realidad Extendida (XR), Sistema de Lenguaje de Diseño (DLS), Interacción Espacial, Diseño Centrado en el Usuario, Diseño Inmersivo.

Abstract (English)

Extended Reality (XR) technologies introduce immersive spatial interactions that challenge the traditional 2D design frameworks used in digital product development. At Philips, the Design Language System (DLS) ensures consistency and usability across platforms but requires adaptation to support XR environments effectively. This thesis employs a user-centered approach to evolve Philips' DLS by identifying missing XR components and translating existing 2D elements into 3D immersive experiences. The research uncovers current limitations and user needs through literature review, market analysis, interviews, and user testing. Participatory methods, including focus groups structured as co-design workshops and card sorting, contribute to the ideation of new interaction components. Components supporting gesture-driven interaction, gaze-based feedback, and 3D spatial navigation were prototyped. *The resulting guideline aims to provide scalable, reusable XR design patterns aligned with Philips' strategic vision, fostering improved usability and consistency in XR applications.*

Keywords:

Extended Reality (XR), Design Language System (DLS), Spatial Interaction, User-Centered Design, Immersive Design

Acknowledgements

As I come to the end of this journey, I want to thank the people who helped me get here and supported me in bringing this thesis to life.

First, I'm really grateful to my supervisor, Angélica De Antonio Jiménez, and my co-supervisor, Robby van Delden, for their guidance, support, and honest feedback throughout the process. I also want to thank Jean-Marc Huijskens for his advice and encouragement during my time at Philips. Their input made a big difference.

To my lovely family and friends, thank you for being there when I was stressed, stuck, or just needed a break. Your support, patience, and care helped me keep going and meant more than I can say.

Glossary

- **AR (Augmented Reality):** Overlays digital content on the real world, enhancing physical environments through a device like a phone or headset.
- **MR (Mixed Reality):** Blends real and virtual environments where physical and digital objects co-exist and interact in real time.
- **VR (Virtual Reality):** A fully immersive digital experience that replaces the real world, typically using a headset and controllers.
- **XR (Extended Reality):** An umbrella term encompassing AR, VR, and MR technologies.
- **DLS (Design Language System):** A structured system of reusable design components, tokens, and principles that ensure visual and functional consistency across a product ecosystem.
- **Image Guided Therapy (IGT):** A medical approach that uses real-time imaging, such as X-ray fluoroscopy, to guide minimally invasive procedures.
- **HCI (Human-Computer Interaction):** The study and design of interactions between people and digital systems, focusing on usability, accessibility, and user experience.
- **UCD (User-Centered Design):** A design methodology that places end-user needs, behaviors, and feedback at the core of the development process.
- **HMD (Head-Mounted Display):** A wearable device, such as a VR or AR headset, that displays digital content in front of the user's eyes.
- **FOV (Field of View):** The extent of the observable environment seen at any moment through a headset, usually measured in degrees.
- **Apple Vision Pro:** Apple's spatial computing headset that supports gaze, gesture, and voice interactions, designed for immersive apps and spatial experiences.
- **HoloLens:** Microsoft's mixed reality headset is designed for enterprise use, enabling spatial interactions through hand gestures and voice commands.
- **ARCore:** Google's development platform for building augmented reality experiences on Android devices using motion tracking, environmental understanding, and light estimation.
- **SUS (System Usability Scale):** A standardized 10-item questionnaire used to evaluate the perceived usability of a system through user ratings.
- **Haptic Feedback:** Physical feedback, such as vibration or force, provided to users to simulate touch or confirm interaction in digital environments.
- **Auditory Feedback:** The use of sound cues (like clicks, tones, or speech) to provide users with feedback on their actions or system status.

Table of Contents

1 Introduction

Extended Reality (XR) has emerged as an important technological advancement in recent years, bringing new challenges and opportunities to the design field. By blending physical and digital environments, XR introduces new ways of interacting with systems, content, and interfaces. However, traditional 2D Design Language Systems (DLS), which define consistent design principles, components, and guidelines, are not inherently suited for immersive spatial experiences. Yet, having a well-defined DLS remains essential for ensuring coherence, usability, and scalability across platforms, especially as digital ecosystems expand into 3D environments.

At Philips, a global health technology company, the DLS plays a key role in maintaining both visual and functional consistency across a wide range of digital products. It helps ensure that all interfaces reflect the brand's identity, usability standards, and clinical reliability. However, XR applications present new challenges: they rely on spatial interaction models, embodied inputs, and environmental awareness that go beyond the capabilities of flat, screen-based design systems. This creates a need to evaluate and adapt Philips' current DLS to support immersive contexts more effectively.

This thesis takes a user-centered approach to transforming Philips' DLS for XR. It focuses on identifying missing elements not currently present in the existing system and exploring how 2D design tokens and components can be meaningfully translated into immersive 3D interactions. The research begins with a thorough literature review and market analysis to understand current XR frameworks and identify their limitations. This is followed by interviews and internal evaluations that examine how the DLS is used within Philips today and what challenges teams face when applying it to XR environments.

To capture insights from both designers and end-users, the study incorporates participatory design methods, including focus groups structured as co-design workshops and card-sorting activities**.** These methods support the ideation phase by surfacing practical needs, validating assumptions, and shaping design priorities for XR adaptation.

Based on these findings, four new interaction components are proposed and prototyped. These include gesture and gaze-based controls, as well as 3D spatial navigation patterns that reflect the expectations and ergonomic needs of XR users. The outcome of the thesis is a comprehensive guide for designing XR interfaces that align with Philips' DLS while extending its functionality into immersive environments.

By bridging the gap between traditional 2D design systems and emerging spatial computing, this thesis contributes to the development of reusable, scalable, and user-friendly XR interaction patterns that support Philips' innovation goals and reflect broader industry trends in digital health and experience design.

2 State of the Art and Market Research

2.1 Overview of XR Design

XR is a term that includes Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR), representing a variety of immersive technologies that combine physical and digital environments. XR enables users to interact with 3D content in real or simulated spaces using a combination of spatial, visual, auditory, and haptic inputs. These technologies are rapidly transforming domains such as healthcare, education, and industrial design by enabling new forms of embodied interaction and contextual awareness. However, as XR interfaces become more complex and adaptive, designers face new challenges in ensuring usability, accessibility, and consistency across devices and platforms. The following sections examine the current state of XR design and interaction frameworks, with an emphasis on theoretical foundations and practical applications in both consumer and professional contexts.

2.2 Theoretical Frameworks and Affordances in XR Design

XR design incorporates both classic human-computer interaction principles and new frameworks that are customized to immersive environments. Recently, researchers have been discussing theoretical models to guide XR interface design to understand how users perceive interactive affordances, also known as cues, on how to act in XR contexts. This section reviews key XR design frameworks and the concept of affordances.

2.2.1 XR Interaction Design Frameworks

Designing for XR interfaces is different than designing 2D ones as unique challenges can arise. To create a seamless UI, the real and virtual content should be integrated into each other without overwhelming the user, and also in a safe way. Todi and Jonker propose a framework for adaptive XR user interfaces defined by five core questions to address this issue [1]:

What? -- *Content selection.* What virtual information or tool should be presented to the user?
How Much? -- *Content density.* How much detail or how many elements should be shown at once?
Where? -- *Spatial placement.* Where in the 3D environment (or on the user's view) should the content appear?
How? -- *Presentation modality.* How should the content be rendered (visual, auditory, haptic) and in what format or style?
When? -- *Timing.* When should the system display or update the content, considering context changes and user tasks?

By considering these 5 questions, this framework provides a structured approach to XR UI design. For example, an AR medical training app might decide *what* virtual guidance to show, *how much* detail based on user expertise, *where* to anchor annotations in the scene, *how* to present them (text, 3D arrows, audio cues), and *when* to reveal or hide them depending on the training step.

Since the XR usage contexts can change rapidly with the user's environment, such adaptive interfaces are seen as critical.

Beyond interface design, the performance and system architecture of XR experiences are critical to effective interaction. XR applications require real-time processing for tracking, rendering, and physics, often on limited hardware, making consistent UI behavior challenging. Huzaifa et al. address this through ILLIXR, an open-source XR testbed that integrates hardware, OS, and algorithms to support end-to-end system evaluation [2]. ILLIXR allows researchers to measure latency, power, and user experience together, reflecting a shift toward domain-specific XR platforms where UI and system design go hand in hand. While technical, this work directly impacts interaction quality, smooth performance, and accurate tracking are essential for effective XR interfaces. In short, XR design spans both content presentation and system-level optimization, both of which shape user experience.

2.2.2 Affordances in XR Environments

The concept of affordances, which can be described as the cues that indicate how to interact with an object or interface, is really important in XR environments as the users face unique interactions that may not be familiar. Vieira *et al.* (2024) conducted a comprehensive literature analysis to understand how affordances are applied in XR and proposed a design space of affordances across eight dimensions. These dimensions characterize different aspects of an XR affordance [3].

Environment Type: In this dimension, VR, AR, and MR contexts are tackled. The majority of studies surveyed were in VR (60%), followed by AR (46.6%), and lastly with MR, which indicates that the research is biased towards VR affordances so far. It is noted that more immersive environments can make tasks feel easier by providing stronger cues.

Modality: This one represents the sensory modality through which the affordance is communicated; it could be visual, auditory, or haptic. Unsurprisingly, *visual* affordances appear in nearly 100% of XR systems due to the graphical nature of XR. However, many systems also incorporate sound or haptic feedback (each in ~50% of surveyed works). Multimodal affordances combining visual, audio, and touch were present in about 13% of cases and have been shown to improve the discovery of interactive elements and user experience. This suggests an opportunity to discover more modalities that are not explored fully, like smell and taste, to improve the experience once technology permits.

Location: This one represents the location where the affordance is presented. For example, on the user's body/device, on a virtual object, or in the surrounding environment. Many XR interfaces use heads-up displays or dashboards anchored to the user's view as a central hub of affordances (buttons, menus), whereas others attach affordances to physical controllers or objects in the scene. Vieira *et al.* found a trend toward multi-location affordances; most XR systems provide interactive cues in multiple places, such as a menu on the controller *and* world-anchored markers, which suggests designers try to distribute interactions in whichever location best fits the context.

Temporal Aspect: This dimension is about whether the affordances are static or dynamic, which means they appear or change in response to context or user actions. All surveyed XR systems involved dynamic affordances that update or animate during interaction. A dynamic example can be a button that highlights when it becomes relevant, or a virtual guide that pops up contextually. These dynamic cues can guide users through complex tasks in ways static cues cannot. However, the challenge is to balance visibility and guidance; poorly timed or overly animated cues might distract or overwhelm users.

Learning Aspect: In this dimension, the relation of affordance to learning is tackled. The affordances are classified by educational intent using Bloom's taxonomy (knowledge, comprehension, application, etc.), since many XR interactions involve learning to use the system itself. Some affordances teach the user a control or feature, while others are designed to let the user learn by doing through exploration. This dimension highlights that XR affordances often serve a pedagogical role, guiding new users to understand novel 3D interaction possibilities.

Task Category: This category tackles the supporting nature of the affordance. For example, an affordance could facilitate navigation, selection/manipulation of virtual objects, communication, or other task types defined in a 3D user interface. Understanding the task category helps designers choose appropriate affordances, like hovering an arrow might afford "go this way" for navigation, whereas a grasp icon might afford "pick up" for manipulation.

Domain Context: This dimension is about the application domain or scenario tasks. Affordances in a gaming context might differ from those in healthcare training or engineering design. Vieira *et al.* note that many XR affordance studies were situated in learning scenarios (education, training simulations), which implies that domain needs strongly influence how affordances are designed.

Affordance Type: This dimension tackles the nature of the affordance itself, particularly whether it is *perceptible* or *hidden*. A perceptible affordance is immediately obvious, for example, a brightly highlighted virtual button, whereas a hidden affordance might not be visible until the user explores or an action reveals it, like a gesture that isn't signposted. Most successful XR interfaces rely on largely perceptible affordances so users aren't lost in the complexity of the virtual world. Hidden affordances can be used for advanced features, but if critical functions are hidden, users may overlook them.

These eight dimensions together create a design space for XR affordances. By mapping XR features across these dimensions, designers can better understand how users interact with the system. Vieira et al. used this framework to analyze current XR design practices and found that most systems rely too much on visual cues. One key insight is that AR is underused, even though it has great potential to use real-world context in meaningful ways. The review also highlights the value of multimodal and multi-location cues, for example, combining visual and audio hints, or placing the same menu icon both in the field of view and on a controller. These strategies help users discover how to interact more easily.

Moreover, designers should also consider users' physical and sensory abilities. Some affordances might work in theory, but be difficult for users with limited fine motor skills or hearing. To sum up, looking at XR design through this multi-dimensional framework helps designers see what's missing, like adding more senses or clearer instructions, and make the system easier and more inclusive to use.

2.2.3 Social and Cognitive Factors in XR Design

In XR design, it's not enough to focus only on interface mechanics. Designers also need to consider how users feel, learn, and interact socially in immersive environments. XR often includes multiple users or virtual agents, and the system design can affect how "present" users feel with others [4]. For example, Shin et al. (2022) found that using fully immersive headsets led to stronger feelings of being together compared to see-through AR displays [5]. It is also shown that the place where the users appear in virtual space is important. For example, placing avatars to the side helped users focus better on tasks, though it reduced eye contact. This means XR design should adapt based on whether the goal is teamwork or individual focus.

Cognitive and learning theories are also important in XR. The study called "A Cognitive Affective Model of Immersive Learning Perspective" identifies six key factors that support learning in XR: *interest, motivation, self-efficacy, embodiment, cognitive load, and self-regulation* [6]. XR apps should keep users motivated, avoid overloading them with information, and provide feedback that builds confidence. Wu et al. (2023) found that these factors helped learners gain and transfer knowledge in XR language learning [6].

Other studies support the idea that combining XR with proven teaching methods improves results. Luo et al. showed that using inquiry-based learning in mixed reality increased engagement and outcomes [7]. Oh et al. (2018) found that in a science museum, users learned better when XR included game-like interactions and full-body gestures, which boosted embodiment [8]. To sum up, a good XR design is more than just visuals; it should also support how people learn, focus, and connect with others.

Overall, theoretical frameworks for XR design now span from *interface-centric models* (layout, adaptation, affordances) to *user-centric models* (social presence, cognitive/affective factors). These frameworks help in designing XR interactions that are intuitive, immersive, and effective. They also highlight open challenges like balancing immersion with usability, or ensuring an XR training tool not only functions technically but also truly facilitates learning and teamwork.

2.3 XR in Healthcare Applications

XR technologies have started to play an important role in the healthcare domain, ranging from professional medical training to patient-facing therapies [9]. This section reviews how XR is being applied in healthcare by analyzing recent studies.

2.3.1 XR for Medical Training and Simulation

Simulation training for healthcare professionals is one of the most common and well-established areas where XR is used. Medical and nursing schools, as well as hospitals, are increasingly adopting VR and AR to train learners in clinical procedures within safe, controlled virtual environments. As Morgan et al. note, XR applications in healthcare "range from surgical simulations and medical training to patient assessments" [9]. For example, VR surgical simulators replicate operating room settings and allow surgeons to practice techniques on virtual anatomy with haptic feedback. These simulations let learners rehearse complex procedures without risk to real patients. Research shows that this kind of VR training can improve surgical performance, leading to shorter procedure times and fewer errors once trainees transition to live cases [10].

AR also plays a key role in medical education. By overlaying visual guidance onto standardized patients, AR headsets can display anatomical structures such as internal organs or tumors directly on the trainee's view during a mock procedure. This reinforces spatial reasoning and supports the affordance of "Where?" by placing information exactly where action is expected to occur. For instance, a student practicing an abdominal exam can see a 3D organ model aligned with the patient's body, helping build anatomical awareness and procedural accuracy.

Beyond formal education, practicing clinicians use XR for ongoing skill development. For instance, some surgeons use patient-specific VR simulations created from real CT or MRI scans as a form of preoperative "warm-up." This is particularly common in fields like orthopedics and neurosurgery, where familiarity with a specific case's anatomy can improve outcomes. Studies report that such personalized rehearsal can reduce complications and increase confidence among experienced surgeons.

Philips' LumiGuide system offers a compelling example of XR-enhanced procedural guidance. LumiGuide provides real-time 3D overlays to support catheter-based procedures, helping doctors visualize devices and vascular structures during complex treatments. While not explicitly tied to Philips' DLS framework, LumiGuide demonstrates how XR is being deployed to enhance training and performance [11].

Remote and shared XR training is also becoming more popular. With AR telepresence, expert doctors can appear virtually in a trainee's space to guide procedures, highlight important areas, or give live feedback. This was especially useful during the COVID-19 pandemic, when travel and in-person training were limited. Many hospitals used XR tools and remote mentoring to keep training going safely.

However, XR training only works well if the system is fast and realistic. If the simulation looks wrong, users may lose trust. This is still a challenge for developers. Platforms like the ILLIXR testbed help solve this by reducing delays and making sure the system responds correctly to the user's actions [2].

Overall, XR provides repeatable, safe, and immersive training experiences that enhance traditional methods. Ongoing research is key to improving these tools and guiding their integration into medical education.

2.3.2 XR for Patient Care and Rehabilitation

XR technologies are increasingly used in clinical settings to support patient comfort and treatment. One major area is therapeutic XR, such as using VR for pain relief and anxiety reduction. Studies have shown that immersing patients in calming virtual environments can significantly reduce discomfort during procedures like wound care or surgery preparation, sometimes lowering the need for medication [12]. VR-guided relaxation has also been shown to ease preoperative anxiety, with evidence supporting XR's value as a non-pharmacological adjunct in healthcare [12].

In mental health, VR exposure therapy is used to treat phobias, PTSD, and anxiety by gradually exposing patients to triggers in a controlled virtual space. Some systems adapt content using biofeedback, making the experience more personalized and effective [13]. Similarly, XR is used in physical rehabilitation to motivate patients through gamified exercises, providing real-time feedback and improving adherence and outcomes, especially in stroke and orthopedic recovery [14].

XR also enhances patient education by visualizing procedures or diagnoses. Tools like 3D heart models or VR walk-throughs of MRI scans improve understanding and reduce anxiety [15]. A notable example comes from Philips' Pediatric Coaching solution for MR imaging. This mobile-based VR/AR tool uses interactive storytelling to help children prepare for MRI scans by simulating the procedure in a gamified, friendly way. This reduces anxiety, lowers sedation rates, and improves scan success, with real-world implementation demonstrating enhanced caregiver and patient satisfaction [16,17].

However, human factors must be considered. As Loizides et al. note, concerns around cybersickness, device usability, and equitable access remain [18]. Addressing these issues is essential to scale XR in clinical practice effectively.

In summary, XR is showing real promise across healthcare, from therapy to education to live clinical support. With thoughtful implementation, it can enhance outcomes, reduce anxiety, and improve the patient experience.

2.3.3 Implementation Challenges and Considerations

While XR has big potential in healthcare, using it in real clinical settings is not always easy. Morgan et al. (2025) explain that for XR to work in hospitals, it must be easy to use, safe, effective, and accepted by both staff and patients [9]. A detailed analysis of the study is shown in Table 1.

TABLE 1

Key Steps and Considerations for Implementing XR in the Medical Field

User Experience	Ensure interfaces are simple and context-appropriate; minimize setup time; provide training for staff and feedback for users.
Hardware Selection	Choose devices that fit the use-case, favor reliable, proven hardware that can be sanitized and handled easily.
Safety and Comfort	Use ergonomic design, allow breaks and real-world view as needed; test for motion sickness and adjust application parameters.
Data and IT	Follow the health data privacy standards; work with hospital IT early to resolve connectivity issues.
Validation	Build an evaluation plan from the start; collect outcome data to demonstrate effectiveness to stakeholders.
Sustainability	Consider the maintenance and updates of the XR content.

In summary, using XR in healthcare isn't just about having advanced technology; it also requires good design, proper training, safety checks, and making sure it follows regulations. The benefits are clear: XR can improve medical training, reduce patient anxiety, and even support new types of therapy. These have been shown in small studies and pilot programs.

However, there are still many challenges. These include technical problems like system delays or data privacy, and human-related concerns like physical comfort, ease of use, and how well XR fits into daily clinical routines. For XR to become a normal part of healthcare, not just an experiment, these issues must be carefully solved.

2.4 Philips' Design Language System

Philips' DLS is a flexible design system created to keep the look of the company's digital health products consistent, usable, and accessible [19]. Since this study focuses on extending the existing DLS, it is important to comprehend its structure. To start with, it includes ready-to-use interface elements, design rules, and examples that help teams build safe and user-friendly tools for clinical use. While it was first made for regular 2D screens, its main ideas, clear layout, accessibility, and adaptable design are still important as XR is an evolving technology. There have been different versions of the DLS over time, but *Filament* is the most recent and updated version used across current projects. Below, the main aspects of DLS are tackled [19].

A key part of the DLS is *accessibility*, which is built into the system from the start. It supports inclusive design for a wide range of users, such as those who are blind, colorblind, visually impaired, deaf, or elderly. The DLS gives clear rules about color contrast, layout, screen reader support, and text scaling. Philips sees accessibility not only as a responsibility, but also to improve the user experience for everyone, including users in temporary or difficult situations.

The DLS also gives guidance for *empty states* when there's no data to show. Instead of leaving screens blank, designers are encouraged to use these moments to help, guide, or encourage users through things like tips, example content, or suggestions. These strategies apply to lists, cards, charts, and error pages to keep users informed and on track.

Another important area of the DLS is how it handles *measurements and annotations* in medical imaging. These tools, like lines, angles, shapes, and labels, are designed to be precise and easy to read, even in complex environments. They follow strict styling rules and can adapt to different screen resolutions and clinical needs, making them reliable tools for diagnosis and communication.

Finally, the DLS supports *responsive and scalable design*. It helps designers build interfaces that work well across different screen sizes and viewing distances, whether on mobile, desktop, or large displays. It includes setup tips for tools like Figma, CSS rules, scaling methods, and design tokens in platforms like React and Angular. It also ensures that accessibility zoom settings don't break the interface.

Overall, the Philips DLS offers a strong base for designing digital healthcare tools. But moving into XR, where users interact with space, depth, gesture, and gaze, means some of these design ideas need to be adapted.

2.5 Philips' Current DLS Approach for XR

Philips has attempted to expand its DLS for use in XR, mainly within its Azurion platform, developed by the Image Guided Therapy (IGT) business [20]. Azurion was created to help clinicians control medical imaging systems directly at the procedure table. Over time, it became a central system for managing digital tools and workflows in hospitals. As part of this progress, Philips started exploring how technologies like AR and MR, especially the Microsoft HoloLens 2, could improve the way medical staff interact with complex tools during procedures.

However, this work on bringing the DLS into XR is only being used in the Azurion project for now. Philips has not created a general XR design guideline that can be used for other simulations or products. The current XR design system is custom-built for Azurion and not meant for wider use at this stage.

In the past, Philips also tried using the Microsoft Mixed Reality Toolkit (MRTK) to build a reusable XR design system. But that project was limited and is now seen as outdated compared to their newer work.

As medical procedures become more complex, with multiple systems and screens, Philips looked at how XR could make things easier. The design team tried to keep the look and feel similar to the 2D version by using the same typography, layouts, and colors. But there were some challenges on the HoloLens 2, dark colors like black or dark gray were hard to see because of how the headset displays light. To fix this, Philips changed some of the color tokens and used special shaders to make sure the interface stayed visible on top of medical images and 3D content.

They tested the HoloLens 2 in clinical simulations to try using gaze, hand gestures, and spatial interaction instead of physical controls. These XR tools were designed to work with both Philips and third-party systems. In a survey at the Society of Interventional Radiology (SIR) conference, more than 75% of doctors who tested the XR demo believed it would affect their work within the next four years.

2.6 Market Research: Comparative Analysis of XR Design Guidelines

In this section, the analysis results of Apple Vision OS, Android XR, Meta's XR, and Microsoft's MRTK guidelines will be discussed, and a comparative approach will be taken [21,22,23,24]. All these systems are the pioneers in the XR world. Apple Vision OS is new and known for its smooth hardware and software working together. Android XR, while newly emerging as a dedicated platform for immersive headsets and still in its early adoption stage, builds on Google's earlier ARCore technology, which has been widely used on Android phones and tablets for mobile AR experiences. Meta's Horizon Platform and Interaction SDK, which powers Quest headsets, brings its own set of interface standards geared toward VR-first and passthrough mixed reality experiences. Microsoft's MRTK, particularly MRTK3, is widely used in HoloLens development and healthcare XR, offering a volumetric design language, gesture systems, and a unit-based layout system tailored for spatial UI prototyping in Unity. This study looks at all four systems to find common ideas and unique features that can help build a strong XR design system for all platforms.

2.6.1 Spatial Layout and Object Placement

Regarding spatial placement, Android's XR guidelines recommend that primary content stay within a 41° field of view to reduce head movement and maintain comfort. On top of that, it is suggested to have a default launch distance of 1.75 meters to enable easy interaction without strain or overreaching. In contrast, Apple Vision Pro introduces the concepts of Shared Space and Full Space, allowing both 2D and 3D elements to coexist. Apple discourages anchoring content to the user's head, advocating instead for a natural and anchored viewing experience that adapts to the physical environment. Meta's guidelines similarly advocate placing UI panels between 1–2 meters from the user, aligned to the natural gaze and avoiding direct anchoring to the head, enhancing environmental context-awareness during interaction. MRTK uses a 1 mm-to-1 unit scale in Unity and supports anchored, world-locked spatial UI layouts with padding and depth rules for ergonomic interaction.

2.6.2 Interactive Elements and Target Sizes

All four platforms emphasize generous sizing for interactive elements to ensure usability. Android suggests a minimum target size of 56dp by 56dp with at least 8dp spacing between items. Apple, leveraging eye-tracking input, recommends a minimum spacing of 60pt between elements to reduce false selections. Meta's guidelines offer 3D collider-based interaction zones with a suggested minimum size of 3.2 cm to accommodate finger poke or raycast interactions. MRTK3 recommends a physical target size of 32mm for button components, with accessible depth cues and collider-based interactions tailored for HoloLens-based precision input. While Android primarily considers gaze and pointer controls, Apple and Meta both focus on natural hand interactions, and MRTK emphasizes flexible multimodal support, including voice, gaze, and articulated hand gestures.

In Table 2, a summary of essential implementation metrics for component sizing, spacing, and input design can be seen for each guideline.

TABLE 2

Comparison of Guidelines based on Essential Metrics

Platform	Recommended Target Size	Spacing Guidelines	Interaction Focus
Android XR	Minimum: 48dp × 48dp Recommended: 56dp × 56dp	Minimum 8dp between elements	Gaze and pointer controls
Apple Vision OS	Minimum 60pt (approx. 2.5cm)	At least 16pt margin around elements	Eye tracking with indirect hand gestures
Meta Horizon OS	Minimum 3.2cm collider hitbox	3D spacing to prevent input collision	Natural hand interactions (poke, raycast)
Microsoft MRTK	32mm physical size	Volumetric padding via Unity anchors	Articulated hands, gaze, voice, controllers

2.6.3 Typography

Typography considerations center around clarity, depth perception, and minimal eye strain. Android recommends fonts of at least 14dp, sentence casing, and adapting text size according to user distance. Apple highlights the importance of placing text in a natural viewing range and maintaining strong contrast ratios. Both discourage long passages of text and motion-bound labels, as they hinder readability and increase cognitive load. Meta applies similar principles, encouraging the use of scalable fonts via Unity's TextMeshPro and curved canvas layouts to reduce distortion and improve legibility in immersive settings. MRTK supports dynamic text scaling and curved canvases, and provides presets for legibility under HoloLens constraints, emphasizing font clarity in varying lighting conditions.

2.6.4 Motion and Animation

Each platform encourages smooth, predictable motion to avoid causing disorientation or motion sickness. Android recommends gradual transitions, fade effects, and pre-animated movement tutorials to help users anticipate UI behavior. Apple also discourages abrupt or large-scale transitions that cut across the user's visual field, preferring soft fade-ins and gentle movement. Meta's system adds state-driven animations (Idle, Hover, Press) and easing curves to create natural transitions. MRTK defines clear animation states and allows for behavior customization through Unity scripting, supporting hover grow/shrink, selection pulses, and feedback tones for immersive and responsive UI behavior.

2.6.5 Interaction Methods and Accessibility

The platforms each support multimodal input, but with different emphasis. Android allows for pointer, gaze, and tap inputs, with hover states and privacy controls especially for eye tracking. Apple reduces physical strain by prioritizing indirect gestures like pinching and tapping in the air, discouraging frequent direct touch interactions. Meta supports ray-based selection, poke gestures, and hand tracking, aiming to replicate embodied, natural interaction. MRTK is highly flexible and supports articulated hand tracking, eye tracking, speech commands, and voice feedback, with modularity built into each component for custom accessibility layers such as dwell interaction, magnification, and input fallback systems.

2.6.6 Immersion and Passthrough

Apple introduces different levels of immersion from partial to full via high-fidelity passthrough and adaptive environments. This lets them choose how much virtual content they see. Android XR, on the other hand, focuses more on panel-based experiences and placing content within a specific field of view (41°), without fully mixing with the real world. Meta enables environmental anchoring through scene understanding APIs, offering full and partial passthrough on Quest devices. MRTK provides scene understanding tools for spatial mapping, hand-anchored content, and blend modes for see-through displays (like HoloLens), enabling environmental alignment in real-time.

2.6.7 Summary

Although Apple, Android, Meta, and MRTK share common design goals like user comfort, spatial efficiency, and accessible interaction, their methods vary significantly. Android leans on traditional UI clarity and device-agnostic design. Apple focuses on blending real and virtual content through natural gestures and environmental anchoring. Meta combines volumetric interaction with immersive 3D feedback, enabling highly embodied user experiences. MRTK contributes a rich set of spatial UI components and a millimeter-based unit system tailored to healthcare and enterprise environments. For a cross-platform XR design system, integrating the best practices from all four Android's structure, Apple's immersion, Meta's spatial depth, and MRTK's ergonomic prototyping can result in a flexible, user-centered framework suitable for healthcare and beyond. Table 3 shows the high-level design guidance across XR platforms.

TABLE 3

XR Design Principles by Platform

Design Aspect	**Android**	**Apple**	**Meta**	**Microsoft**
Interaction Design	Min 56dp size, 8dp spacing	60pt spacing, optimized for eye tracking	Min 3.2cm button size, 3D collider feedback	Min $32mm^3$ pressable volume; supports tap, pinch, dwell
Typography & Legibility	14dp+, sentence case, high contrast	Eye-level, high contrast, avoid motion-based text	Curved canvases, dynamic text scaling via TextMeshPro	Scalable, high-contrast text optimized for mixed environments
Motion & Transitions	Smooth fades, no abrupt changes	Gentle animations, avoid large object motion	Eased transitions, idle/hover/press states, no jumpy UI	State-driven (focus, press, disabled) with light motion feedback
Spatial Placement	41° FOV, 1.75m distance	Anchored, not head-bound;Shared/Full Space modes	World-locked UI within 1–2m, gaze-centered panels	World or surface-anchored UI; ideal range ~1–2m
Accessibility & Hover States	Gaze + pointer hover, eye	Low-effort gestures, indirect interactions	Ray/poke/grab, passthrough accessibility settings	Gaze focus, one-handed controls, high-contrast themes

	tracking privacy			
Passthrough & Environment	Limited blending, panel-focused	Immersive passthrough spectrum	Full/partial passthrough with environment-aware UI	Spatial mapping, occlusion-aware content on HoloLens
Gestural Interactions	Pointer or gaze emphasis	Indirect pinch, tap, minimal fatigue	Hand tracking with spatial feedback, controller fallback	Built-in gesture set (grab, scroll, touch) with contextual behaviors

2.7 Summary and Research Gaps

To sum up, Chapter 2 covered the theoretical part of XR interaction design and explored the state of XR applications in healthcare. Key points include the frameworks for XR design that address *what* content to show, *where/when* to show it, and *how* to present it for optimal user experience [1]. Besides, user-centered theories (like social presence and immersive learning models) inform how XR can engage users at a psychological level [4,6]. Another key point was XR in Healthcare, it is seen that XR has a great potential to play a critical role in this domain, and the simulations can enhance medical education and skills practice, while immersive therapies can reduce patient pain and anxiety [12]. After reviewing the literature, several research gaps have emerged at the intersection of XR design and healthcare.

Gap 1: Comprehensive Design Guidelines for XR Interfaces

Despite new frameworks, there is no set of design guidelines for creating effective XR user interfaces across different contexts. Designers lack concrete best practices for implementing affordances or adaptive UI elements optimally. For instance, Vieira *et al.* identify dimensions of affordances [3], but translating those into design rules like how to choose the right modality or temporal cue for a given task remains an open challenge. More research needs to be done to create actionable guidelines.

Gap 2: Contextual Adaptation in XR

Adaptive XR interfaces are still largely conceptual. Todi's framework proposes adapting content and presentation to context [1], but few implemented systems intelligently adjust to a user's environment, skill level, or biofeedback in real time. In healthcare, especially, the ability to personalize XR experiences (e.g., simplifying the interface for a novice vs. expert surgeon, or adjusting therapy difficulty to a patient's progress) could greatly enhance effectiveness. Developing computational methods and algorithms for context-aware XR UI adaptation and evaluating their impact is a ripe area for exploration.

Gap 3: User-Centered Design and Co-Creation for XR Health Applications

Many XR healthcare solutions have been technology-driven rather than co-designed with end users. The literature review pointed to the need for participatory design: healthcare XR tools must fit the users' workflows and address real user needs. However, systematic methodologies for involving clinicians and patients in XR design are not well documented. There is a gap in understanding how co-design insights can be effectively incorporated into the XR development lifecycle.

Gap 4: Integration into Clinical Practice

Bridging the "last mile" from XR prototype to routine clinical use is poorly understood. As Morgan *et al.* highlight, there is scant guidance on how to deploy XR at the bedside or clinic floor [9]. Issues such as device sterilization protocols, IT support, staff training programs, and cost justification models are not thoroughly addressed in current research. Identifying strategies to integrate XR solutions into existing healthcare systems (both technically and organizationally) is a gap that must be filled for XR to move beyond demonstrations. This includes understanding change management: what factors lead to sustained adoption of XR by healthcare professionals?

Gap 5: Evaluation of Efficacy and Outcomes

While many XR applications show promise in short-term, controlled evaluations, there is a lack of long-term and large-scale studies confirming their efficacy. For example, does VR training translate to measurable improvements in patient care quality over time? Do AR-assisted surgeries consistently improve outcomes across many cases? And critically, what is the impact on patient health and well-being? The literature reveals a need for more clinical trials and longitudinal studies of XR interventions. Closing this gap will require collaboration between technologists, clinicians, and researchers in medical science to design studies that capture not just usability, but health outcomes and cost-effectiveness.

Gap 6: Technical Limitations and Human Factors

Current XR technology still faces technical limitations like lower resolution or battery life, and human factor issues such as VR sickness that constrain what is possible in healthcare scenarios. Research is needed on how to mitigate these limitations, either through improved hardware/software or through creative design. Furthermore, ensuring accessibility of XR for users of varying abilities (vision, hearing, mobility impairments) is an often overlooked aspect; therefore, an inclusive design gap exists in XR that is crucial in the context of healthcare.

These gaps form the base for this research. By addressing elements of these issues, particularly adaptive design (Gap 2), co-design methodology (Gap 3), and evaluating XR in a realistic healthcare training context (Gap 5) this thesis aims to contribute knowledge that brings XR interaction design and healthcare practice closer together. In the next chapters, the user-centered and participatory methods that are used to start creating an XR-oriented DLS will be tackled.

3 User-Centered Methods: Interviews and Usability Tests

To obtain a comprehensive understanding of the current state and usage of the DLS within Philips, two semi-structured interviews were conducted with key stakeholders: the head of design, responsible for defining the DLS strategy, and a design system head responsible for the implementation side. This method provided perspectives on how the DLS is perceived within the company and gave an idea of how to extend the DLS to XR environments.

In addition to these expert insights, two different types of user tests with two different XR simulations currently in use within the company were conducted, one using the think-aloud method with a smaller group, where users talk during the task and observational techniques, and the other with a post-session survey with a larger sample set. This method provided a point of view on their current simulations and what the problems are. Besides that, it also provided a perspective on how DLS is experienced in real-world design scenarios. Before each session participants were informed by information sheets and provided by consent form which can be seen in Appendices A and B. This procedure was assessed and provided a positive assessment which is registered under Approval nr 250244 of the ethical committee of Computer & Information Sciences (CIS) from the EEMCS faculty of the University of Twente.

3.1 Interviews

3.1.1 Methodology

Two semi-structured interviews were conducted with senior members of the Philips design system team. Each session lasted between 20 and 30 minutes; one of them was online, whereas the other one was in-person. Both occurred in February 2025. The interviews followed a flexible guide that had different sets of questions based on the role of the participants. The data was collected via transcribing the audio recordings. Notes were also taken during the sessions to capture contextual observations and follow-up points. Both transcripts were anonymized and analyzed to extract key themes related to XR adaptation.

The data was analyzed based on the question categories, and the categories were used to structure the results and the discussion section that follows this methodology. The interview guides that were used with the whole set of questions can be found in Appendices C and D, and a general overview of the two interview guides is shown below in Table 4.

TABLE 4

Overview of Interview Guidelines

Categories	Interview 1: DLS Strategic/Design Lead	Interview 2: DLS Technical/Implementation Lead
Focus Area	Conceptual design principles, visual design tools, DLS governance	Technical integration, framework compatibility, component testing, scalability

XR Emphasis	Conceptual feasibility of translating DLS to XR	Technical feasibility and implementation barriers in XR
Warm-up	Role and experience in defining DLS principles	Role and experience in framework/tool integration
Main Sections	- DLS structure & core values - Design & iteration workflow - Testing and design feedback - Opinions on XR design challenges	- Toolkit usage & adoption - Prototyping workflow - Accessibility practices - Testing/feedback channels - System versioning and updates - XR implementation challenges
Wrap-up	Referrals, final thoughts on XR	Referrals, follow-up, and XR support opportunities

3.1.2 Interview Results

The results of the interviews provided a comprehensive view of how Philips uses and develops DLS, and insights on how to transform 2D structure to 3D. The findings are grouped into four categories: structure and implementation, tools and workflows, accessibility and testing, and XR considerations. Full transcript can be seen in Appendices E and F.

3.1.2.1 Structure and Implementation of the DLS

Philips' DLS follows a three-layer architecture: foundations (e.g., colors, typography, spacing), components (e.g., buttons, cards), and templates or architectures (screen-level layouts and workflows). This modular structure enables reuse across platforms and helps maintain design consistency. While the DLS is widely used, full adoption is still a challenge, particularly among teams in the maintenance phase of a product life cycle. One of the main problems is persuading teams to invest the effort needed for migration, especially when their current systems are stable and functional.

3.1.2.2 Tools and Workflows

The design process is highly collaborative, and tool driven. Figma is the primary design tool, with Token Studio used to manage design tokens (though it may soon be replaced). Developers use GitHub for managing component development, while Storybook and Chromatic support component documentation and visual regression testing.

The workflow typically begins when a business identifies a need or requests a component update. UX designers start the initial designs in Figma, which then go through iterations with developers. The refined component is documented, tested, and implemented across multiple frameworks like React, iOS, Android, and Angular.

3.1.2.3 Accessibility and Testing

Accessibility is a key quality metric and is primarily handled through manual testing and plugin-assisted reviews in Figma. Philips aims to meet WCAG AA compliance as a standard, with AAA targeted when possible, depending on the context. While foundational accessibility (e.g., contrast, font size) is addressed during design, technical accessibility varies by platform and is tested accordingly. A team member with accessibility expertise monitors this across technologies. Usability testing, however, is generally performed at the product level by the respective business units, not directly within the DLS team. Feedback from these sessions is used to refine or fix components when needed.

3.1.2.4 Extending the DLS to XR

When asked about adapting the DLS to XR, both participants expressed that core UI elements like buttons, menus, and text remain applicable in 3D spaces. However, they also acknowledged that XR introduces specific challenges. Navigation and spatial interaction need to be considered, as traditional menus or scroll interactions may not translate well. In clinical contexts, where interfaces are task-critical, transitioning to XR requires careful design. For example, showing all critical information while preserving spatial usability is more complex than in consumer applications.

Philips previously developed an older XR-focused DLS, as mentioned in the previous chapter, which may provide useful reference material for current or future development. Participants emphasized that XR components shouldn’t be a direct 1:1 translation of 2D UI elements. Instead, new interaction models, including gesture input, gaze selection, spatial anchoring, and adaptive layout will need to be explored. Additionally, accessibility in XR raises new concerns, such as visual clarity in immersive environments and support for alternative input methods.

3.1.2.5 Interview Discussion

The interviews gave me a good understanding of how Philips’ DLS is used, how it is structured, and how different teams work with it. I learned which parts of the system are used most often, which ones are important for accessibility, and how people work together when designing with it. This helped me see which parts of the DLS can be used in XR and which parts need to be changed or added for 3D environments. I had learnt which components are essential, reusable, or only useful in certain situations. The goal is to design XR versions that still follow Philips’ style but work better in 3D. The interviews also showed that some features needed for XR, gesture controls and 3D navigation, are missing from the current DLS. Also, common 2D design patterns, like fixed layouts, don't work well in XR. That’s why it’s important to create new, more flexible, and user-friendly components for these environments. These insights helped guide my design choices and make sure the XR components I create are both useful and consistent with the rest of Philips’ design system.

3.2 User Tests

3.2.1 Methodology

To evaluate user interaction and usability within XR healthcare simulations, two complementary studies were conducted between March and May 2025, each using a different XR scenario to explore current challenges and limitations. Both simulations featured virtual representations of Philips medical devices but offered different levels of interactivity. Participants were Philips employees with varying backgrounds and experience in VR.

Study 1 included three participants and was conducted as a focused usability test for a specific team. It involved a longer session format using think-aloud protocols and detailed observation. In contrast, Study 2 involved 25 participants during an internal open day, where a shorter post-demo survey made it possible to gather feedback from a larger group in a limited time. This approach allowed for both in-depth insights and broader user perspectives.

3.2.1.1 Study 1: Simulation with Think-Aloud Protocol

The first study involved 3 usability test sessions where participants interacted with an MR simulation using the Apple Vision Pro. The simulation represented a medical environment where various virtual medical devices were displayed, and one of them could be interacted with. Participants were placed in a controlled, room-like virtual setting that included labeled equipment, floating navigation cues in the form of stars, and spatial audio prompts. Although the exact context cannot be disclosed for privacy reasons, users were asked to complete a set of guided tasks. These included following the visual stars to reach different virtual machines, reading brief information panels near each one, and interacting with one device by pressing a virtual button. In addition to the visual indicators, spatial audio cues reinforced direction and task progress, offering layered feedback for navigation.

Throughout each session, the think-aloud protocol was used, with participants encouraged to verbalize their thoughts while navigating the experience. Observational notes were also taken, focusing on behavior, moments of confusion, and feedback on interface elements, particularly the interactive button, the star-based guidance indicators, and the use of spatial sound.

3.2.1.2 Study 2: Interactive Simulation with Post-Demo Survey

In the second study, a more interactive MR simulation was conducted using the Meta Quest headset without physical controllers. Instead, the system featured a virtual floating controller that appeared in the scene and could be operated through hand tracking. Users were able to see virtual silhouettes of their hands, which allowed them to interact with the controller and manipulate a virtual Philips medical machine (see Fig. 1). This included performing simple actions like repositioning the virtual machine and exploring its components, buttons, knobs, and handles, to better understand its function. The simulation served as an early prototype for a potential training tool aimed at familiarizing users with complex medical equipment in a hands-on, immersive way and was not created by the author but was an existing internal prototype developed by Philips and demonstrated during an open company day.

This study included a larger group of 25 participants. Unlike the first study, participants were not asked to think-aloud. Instead, they completed a short post-experience questionnaire to provide both quantitative and qualitative feedback:

1. Overall experience (rated from 1 to 5, with optional open-ended comments)
2. Ease of interaction
3. Suggestions for improving realism and engagement

Fig. 1. Zenition 90-medical device displayed in simulation

Although the feedback method was simpler, this study enabled the collection of insights from a broader sample. Observational notes were also recorded, focusing on how well hand tracking worked, how intuitive and responsive the virtual controller felt, how clearly participants understood the spatial layout, and whether they expected tactile feedback during interaction. The absence of physical controllers provided a more natural interaction experience, but also raised expectations for responsive and intuitive system behavior.

Fig.2,3. Study 2 demo session

By combining these two studies, a good mix of information was obtained: the first one gives detailed thoughts from users and lets me understand from the user’s point of view, while the second shows general trends, provides quantitative data, and how people react to a more hands-on XR prototype. This way, a comprehensive view is achieved on how users interact with these simulations, and a user-centered approach was able to improve the current systems and take them as an example for future simulations.

3.2.2 Results of Study 1

Based on the observational feedback and the think-aloud technique, the first key point was regarding button interactions; the participants tried to press elements that were not buttons, but labels or inactive elements. Therefore, it is detected that the UI of the buttons or the labels is unclear and confusing. Similarly to buttons, the pinch-based selection, which is the default type of Apple Vision Pro, was also unclear to the users. They tried to press or apply excessive pressure instead of the pinch gesture, although it was introduced at the onboarding stage, which also shows that onboarding was unclear. One of the participants even skipped the onboarding and tried to interact with the system immediately. The calibration phase was also confusing, as users found it too long and overwhelming.

Floating stars are implemented as navigation elements, which helped the participants, but 1 of 3 participants did not understand their purpos. Another hint system was audible cues to navigate users, but it was inaudible due to speaker-based output rather than headset audio. Participants could not determine the directionality of sounds, which reduced the clarity of the simulation.

The reality of the simulation was rated good, but the transparency was confusing, as users had a hard time differentiating between the real objects and the virtual ones. Another thing that disturbed them was the position of the virtual objects, as they were too close to the users, which disrupted the sense of personal space. A summary of participant feedback is given in Table 5.

TABLE 5

Feedback Summary of Study 1

Aspect	Participant 1	Participant 2	Participant 3
Navigation	Smooth and intuitive	Natural and fluid	Skipped onboarding; started exploring prematurely
Button Clarity	Purpose of buttons unclear (“Proceed” button), tried to press non-interactive labels	Unsure whether to "tap" or "touch" preferred "touch"	Tried to press non-interactive labels; unclear button interactions
Gesture Interaction	–	–	Pinching unclear; pressed too long; struggled to enter simulation
Sound Design	Soundscape enjoyable, headset preferred	Could not hear sounds	Could not hear sounds
Graphics& Environment	Well-designed and visually appealing	Found the virtual machine engaging and believable	Transparency was disorienting; couldn't distinguish real vs. virtual table
Guidance	Stars helpful once understood	Needed additional onboarding support	Did not recognize stars as navigation guidance initially
System Understanding	Couldn’t understand what to interact with	–	Asked whether the user could interact at all

Positional Feedback	–	–	Found character positioning unnatural (appeared too close)

3.2.3 Study 1 Discussion and Design Suggestions

Gestural interaction could feel more intuitive by replacing or complementing the pinch gesture with something simpler, like tapping or hovering, but as it is the default feature of Apple Vision Pro, the labels can be made clearer, indicating that they are not interactive elements. Adding quick visual or haptic feedback or showing a short tutorial at the start might help users feel more confident using gestures.

The onboarding flow could be made clearer and harder to skip, so users don't miss important setup steps. A conversational agent can be placed instead of a text-based introduction. Calibration could also be automated, with brief prompts to guide users through it without breaking immersion.

To improve navigation, the star markers could be introduced with a quick animation and sound cue, so users immediately understand what they're for. If someone seems lost or idle, subtle hints like arrows or fading trails could lead them in the right direction.

Sound cues could be routed through the headset by default to support better spatial awareness. Pairing important sounds with small visual effects like glows or highlights might make sure users don't miss them, especially in more complex scenes.

Since some users found transparency or visual details a bit confusing. Making surrounding elements more visually distinct by using outlines, glow, or contrast could help separate what's part of the simulation and what's not.

3.2.4 Results of Study 2

The second usability study included 25 participants who tested a more interactive simulation using an AR controller with buttons and knobs. This prototype was developed to showcase a Philips product and its potential use for future medical training events. Unlike the first study, this version allowed users to directly manipulate parts of a virtual medical machine, including moving it, pressing buttons, and exploring different controls. The simulation was accessed through hand tracking without physical controllers, where a floating virtual controller and visible hand silhouettes were used for interaction.

After the simulation, participants were asked to respond to three short questions. Regarding overall experience, most participants rated the simulation positively (4 or 5 out of 5), highlighting the system's immersiveness, realism, and visual quality. Some, however, mentioned that setup time, lack of onboarding, and limited feedback reduced clarity, especially for first-time XR users.

Regarding ease of use, participants appreciated the natural feel of hand tracking, but several expressed confusion about how to operate certain elements, such as grabbing knobs or pressing buttons, with some unsure if interaction was even possible.

Realism and engagement were also rated highly, with many users noting that the machine looked and behaved convincingly. However, suggestions for improvement included adding haptic or audio feedback, increasing interactivity, making text more legible, and incorporating a narrative or training scenario for context.

While the think-aloud protocol was not used, these responses, combined with observational notes, provided valuable insights into how users interacted with the system. Most participants said the controller helped them understand the machine better, but some had trouble figuring out what each button or knob did. They expected a more guided explanation of what was happening during the interaction. Hand tracking and system responsiveness were generally well-received, though a few people mentioned that the controls could feel more responsive or clearer in their function. Some also pointed out that they wanted more tactile or visual feedback when interacting with the machine. Spatial understanding was strong overall, and participants could follow the interface and understand how to position and move the device. However, several expected clearer confirmation after pressing a button or completing a task, suggesting a need for stronger feedback cues.

A summary of participant feedback is provided in the following table (see Table 6), and the detailed notes for each participant are shown in Appendix G.

TABLE 6

Feedback Summary of Study 2

Question Category	**Feedback Summary**
Overall Experience	Most users rated the experience positively (scores of 4 or 5), highlighting the immersive quality, visual realism, and innovation. Some participants noted setup delays, initial confusion, and lack of onboarding as detractors.
Ease of Use	Hand tracking was generally smooth and intuitive. However, users were occasionally unsure how to operate the virtual knobs and buttons. Several expected clearer interaction guidance and more consistent responsiveness.
Realism & Engagement	The virtual machine was seen as realistic and well-designed. Many users desired haptic or audio feedback, clearer visual cues, or contextual elements like a storyline or task to increase engagement.

3.2.5 Study 2 Discussion and Design Suggestions

The second study showed that participants generally enjoyed the experience. Most of them gave high scores, 4 or 5 out of 5, saying it felt engaging and realistic. Only a few participants (3 out of 25) gave lower scores, and that was mostly because of issues with using the system easily.

One of the biggest problems was ease of use. Users had trouble with hand tracking, grabbing objects, and pressing buttons accurately. Some reported delays or that the system didn't respond the way they expected. To improve this, the interactions could be made more responsive, and clearer visual or tactile feedback (like a light vibration, click sound, or glow) should be added after each action so users know it worked, since no controllers are involved. If it is decided to use controllers by the design team, then the tactile feedback could be vibrations directly.

Another key issue was onboarding. Some users felt a bit lost at the beginning, not knowing what to do with the controller or how to interact with the virtual machine. Adding a short, step-by-step training or guided demo at the start would help users understand the controls and feel more confident.

Despite these challenges, the realism of the simulation was rated highly. Participants liked the detailed visuals and the immersive feel. A few users suggested that the experience would feel even more real if it included a short storyline or had roles like a patient or nurse. This kind of narrative could help users better understand the context and make the training more memorable.

To sum up, users appreciated the interactive design and realism of the system but wanted better onboarding, more responsive controls, and a bit more guidance. Adding these elements would make the simulation easier to use, more immersive, and better suited for training.

4 Participatory Methods: Focus Group & Co-Design Workshop and Card Sorting

After understanding what DLS is and how they use it, participatory methods were conducted. The main reason behind this is to obtain a full understanding of how to design interaction components that are built upon a user-centered approach. I conducted two different participatory methods, involving both end users and internal design stakeholders in the research process. The first method was a series of focus group sessions, including a co-design workshop with end users, and the second one was a hybrid card sorting activity with Philips designers and developers. This approach enabled me to see the whole picture with the help of the people who use XR technologies and from those who design and implement them.

The co-design workshops were structured as an open-ended, exploratory space for users to reflect on their previous VR experiences and imagine new interaction patterns without any constraints. This flexibility allowed for the discovery of interaction needs that do not currently exist in 2D DLS. On the other hand, the card sorting activity focused on evaluating how existing elements of Philips' DLS are interpreted when translated into XR environments. The participants were asked to sort existing components into predetermined groups and make their own if they did not exist. This approach allowed me to understand design knowledge and platform assumptions held within the organization.

Together, these two methods, which will be elaborated in the following sections, provide a comprehensive view: the focus group helped identify novel interaction patterns and missing components that need to be designed from scratch for XR, while the card sorting revealed which DLS elements could be adapted, discarded, or redefined. Again, during conducting these sessions, consent forms and participant information sheets were provided (see Appendices A and B). This procedure was integrated into the same assessment which was provided a positive assessment and was registered under Approval nr 250244 of the ethical committee of Computer & Information Sciences (CIS) from the EEMCS faculty of the University of Twente.

4.1 Focus Group & Co-Design Workshop with End Users

The first method was focus group sessions that were combined with co-design workshops. The reason for this structure is to obtain in-depth qualitative feedback on participants' experiences with VR and to collaboratively ideate new interaction components. The structure encouraged participants to evaluate existing interactions, brainstorm alternatives, and sketch conceptual components using guidance based on spatial, context-aware, and embodied interaction paradigms.

To investigate user preferences and ideate interaction elements for XR, I conducted three sessions. These sessions served both as a means of gathering qualitative insights into user experiences with XR and as a creative space to co-develop interaction concepts that better suit user needs.

4.1.1 Methodology

4.1.1.1 Participants & Recruitment

A total of 13 participants took part across three sessions in the following pattern: 4-4-5 for each session, respectively. All conducted in April 2025. Two of the sessions were conducted in person, and one was held online to include remote participants. The ages of the participants ranged from 23 to 29, and they represented five different nationalities. The participant group included both university students and young professionals from diverse backgrounds. Participants were recruited through personal networks, including university WhatsApp group chats, private messages, and outreach to young professionals in related fields. Participation was voluntary, and those who agreed were scheduled using *LettuceMeet*, an online scheduling tool that enabled optimal time matching.

A test session was conducted with 2 people before the main ones to obtain some feedback, and minor adjustments were made regarding the session structure, ensuring a smoother participant experience.

4.1.1.2 Workshop Structure

Each session followed a structured, two-part format combining the focus group part, including discussion and evaluation, and the co-design part, including the creative ideation and reflection part. Overall, each session lasted approximately 45 minutes. A notetaker was also involved in face-to-face sessions to take notes and observe the environment while the researcher conducted the session. The presentation that was used can be found in Appendix H.

Part 1: Experience Sharing and Interaction Evaluation

The sessions began with an informal icebreaker discussion, where participants were invited to share their favorite or most unusual VR experiences. This helped to have a comfortable setting to open a dialogue.

After this, participants were asked to evaluate the interactions they had encountered in VR. They reflected on which interactions felt intuitive or frustrating based on the examples that they gave in the previous round. After this discussion, participants were asked to imagine themselves in a VR environment and do the following tasks;

- Selecting an item
- Opening a menu
- Going back or undoing an action
- Getting help or feedback

An example VR environment was not given to create a flexible atmosphere so that every detail can be designed by the imagination of the participant. They were allowed to use pen and paper at this stage. Participants were then asked to imagine improvements or alternatives to these tasks. A minute was given for them to complete the brainstorming, and they were asked to share their experience in a round robin structure, where each participant is allowed to talk in the same order. This included suggesting new gestures or contextual models that could enhance or replace existing interaction methods.

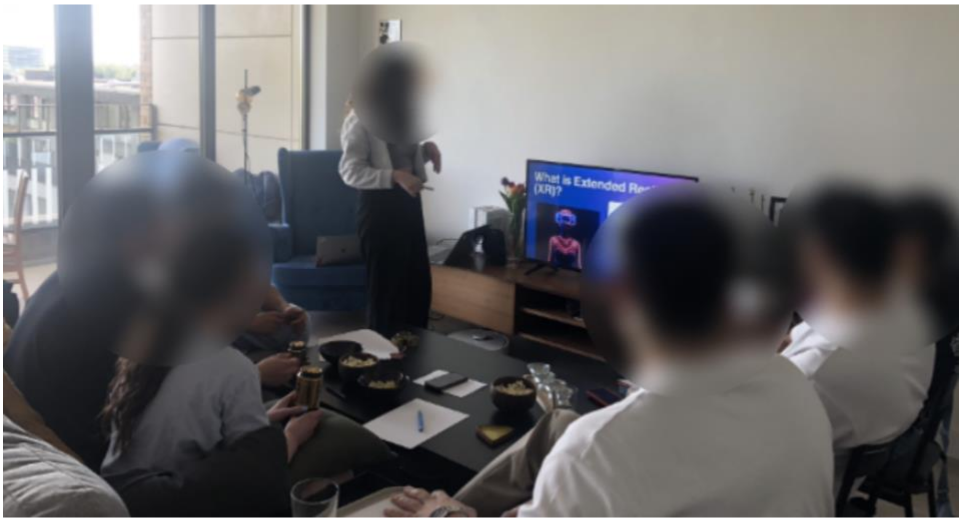


Fig.4. Focus group 3

Fig.5. Focus group 3 co-design phase

Part 2: Introduction to XR Interaction Categories & Co-Design Activity

In the second half of the session, participants were introduced to three categories of XR-native interaction components. These categories were chosen based on current platform guidelines and existing HCI literature, as discussed in the market research section. Apple Vision Pro's Human Interface Guidelines highlight gesture-driven input and gaze-based selection as foundational interaction models [21]. These reflect not just design preferences but also align with established interaction paradigms in XR environments.

To structure the discussion and ideation activities, three specific paradigms were used: spatial, context-aware, and embodied interaction. *Spatial interaction* focuses on how components are positioned and interacted with in 3D space, for example, placing a menu near a physical object or anchored to the user's body. *Context-aware interaction* involves components that adapt to the environment or user behavior, such as showing a help panel when a user pauses. *Embodied interaction* relies on body-based inputs, including gestures or gaze, and has been described as a core element of immersive systems by both researchers and platforms [25,26,27,28].

The goal was to guide participants to reflect on these interaction types while brainstorming and sketching new components. This helped ensure that the design ideas were grounded in real use cases and supported by existing theory.

Participants then engaged in a co-design activity, where they were asked to invent their interaction component inspired by these categories. They were allowed to choose one of these categories or create their own by combining them. They were asked to use pen and paper to foster creativity, and they individually or in groups brainstormed ideas, named their components, described what they would do, how they would work, how the user would interact with it, and which category they chose or created. Sketching was encouraged to visualize their concepts.

Part 3: Reflection

Each session concluded with a group reflection, where participants shared their designs and discussed their preferred interaction types and the biggest gaps or frustrations they had noticed in current XR systems.

Fig.6,7. Focus group 3 reflection phase

4.1.2 Results of Focus Group Sessions

In this section, the results of the focus group sessions, which incorporated co-design workshops, are analyzed. In each session, there were 4-5 participants, and in total,13 participants were recruited. The participants had different levels of XR experience, from advanced users to total beginners who had little experience in various XR environments. Some experience examples are meditative VR, a physiotherapy game, a war journalism simulation, and an escape room.

Thematic analysis was applied using the recordings and the transcripts, which resulted in 5 dominant themes, which are elaborated below.

4.1.2.1 Embodied Interaction

In all sessions, the importance of natural, body-based gestures in XR was highlighted. 85% of the participants mentioned that they would prefer using intuitive, embodied gestures such as grabbing, pinching, pointing, or swiping over reliance on external controllers.

In the second part, most of the designed interaction elements were activated by embodied gestures or moves. One group created a flying tour using their arms like birds to explore surroundings, lifting the left or right arm to turn to the respective side, pinching to zoom to see the details, and a voice-based menu. Another team had an underwater simulation where the user needs to do the swimming moves to be able to move around, where faster strokes increase speed, having wearables to measure oxygen & stress levels, and voice checks from the boat that can be received by the headset if the user wants to end the simulation.

Other embodied gestures that stood out were hugging oneself for help, hitting the wrist or chest to open different menus, rotating fingers for music volume, or a personal dashboard anchored to the user's wrist for tools/status, or pressing the wrist for the menu.

Overall, gesture control is highly appreciated, especially when it mirrors real-world physicality (e.g., grabbing, pinching, or pushing). Participants find it intuitive and immersive, but only if it is simple, natural, accurate, and responsive, and not tiring over time.

> "I like being able to just reach out and grab things. It feels way more natural than using some controller." – FG3_P2

> "It's about feeling like you're actually touching or doing something, not pressing buttons in the air." – FG3_P3

The feedback highlights the importance of a DLS that has embodied gesture tokens that are standardized, ergonomic, and with adaptable motion patterns. With these tokens, it can be assured that the learning difficulty can be reduced, and consistency can be improved.

4.1.2.1.1 DLS Implication:

- Specify reusable gesture primitives, such as swipe, point, pinch, and grasp.
- Design components with gesture hooks (e.g., Help, Undo, Menu) that can be triggered via multiple input methods.
- Provide alternatives for users with limited movement.
- Introduce embodied menu tokens components that attach to body zones and appear based on spatial gestures.

4.1.2.2 Context-Aware Interfaces

The second category is based on the context-aware interfaces. It is mentioned that situational awareness and adaptability are a demand in XR environments. A majority of participants, around 70%, said they prefer interfaces that adjust their complexity or visibility based on the user's context, like task intensity, motion state, or social setting. One of the examples is looking at an object for more than 10 seconds triggers information that is designed in the second part as an interaction element.

> "If you look at something for 10 seconds, it shows information about it. When you move again, the popup disappears." – FG3_P3

A similar scenario designed by the participants was having a menu that opens or confirms selection after 2–3 seconds of fixation. Another example is based on a classroom environment, if the student does not interact with the components for some time, a pop-up window appears in front of them, encouraging them to interact with the teacher and increase the engagement. It could be a question related to the flow of the lesson. They also mentioned that they don't want to be overwhelmed by the components, so having context-aware interfaces could be helpful.

> "When I'm doing something active, I don't want my view to be crowded with extra buttons or menus." – FG2_P1

4.1.2.2.1 DLS Implication:

- Components should monitor dwell time to reveal or hide content based on user attention.
- Introduce quiet/passive modes where interaction components fade or remain dormant.
- UI layout should be aware of spatial boundaries and dynamically adapt to movement and user position.
- Allow app developers to mark elements as essential, helpful, or optional to control how they appear in different scenarios.

4.1.2.3 Accessibility and Customization

Another category was accessibility, particularly in the second focus group, but also brought up in other sessions as a desirable feature to increase inclusivity. Participants also discussed and suggested customizable input types, including voice commands, eye tracking, haptic cues, and alternative gestures. 62% of them explicitly suggested personalization settings that let them customize interactions based on their physical or cognitive preferences.

> "Different people interact in different ways, why not let them choose?" –FG2_P2
> "I want to pick what works best for me, sometimes I'm tired and prefer voice." – FG2_P4.

Another feedback was replacing text-heavy onboarding with an avatar-based assistant that speaks or gestures.

> "I don't want to read instructions, just have someone show me like in a game." – FG1_P1.

4.1.2.3.1 DLS Implication:

- Support a gesture library where users can enable/disable or reassign specific motions.
- The system should allow seamless switching between modes.
- Use progressive disclosure and tiered interfaces.
- Implement narrated, animated onboarding agents instead of modal text screens.

4.1.2.4 Modular & Minimal UI

This category is highlighted in all sessions, claiming that participants prefer lightweight, modular interfaces that support spatial adaptability and prevent overwhelming situations. The term "floating" or "hovering" UI appeared repeatedly, with several participants advocating for spatially anchored menus that only appear when needed. The users also suggested being able to hide or pin the required components related to the task, which also enables personalization and aligns with the previous category.

> "Let me decide what tools stay on-screen. I don't need everything all the time." – FG2_P3

It is seen that participants prefer dynamic layouts that include elements repositioned based on gaze, movement, or user-defined zones. They also mentioned that if a component is placed somewhere they can’t see, there should be an indication so that they don’t get overwhelmed. Radial menus, floating toolbars, and pop-in widgets were frequently mentioned as useful components.

4.1.2.4.1 DLS Implication:

- Add tokens for responsive menu behavior with proximity activation and gaze sensitivity.
- Define tokens with behaviors like "auto-hide", "auto-pin".
- Include layout-independent tokens that define component behavior, not just visual styling.
- Support adaptive logic for when, where, and how components appear.

4.1.2.5 Global Interaction Patterns

Participants expressed frustration over inconsistency across XR apps in terms of gestures, navigation, and feedback. Approximately 75% of users noted that it is hard to adapt to new XR environments because “nothing works the same way twice.” For example, users expect a consistent way for basic interactions like cancel, go back, select, or scroll.

> “Undo could be a simple backward motion.” – FG3_P3

There was high demand for a unified grammar of gestures and feedback, similar to how iOS or Android maintains system-wide behaviors.

> “I just want one back gesture that works everywhere.” – FG2_P1

4.1.2.5.1 DLS Implication:

- Define platform-wide interaction rules for common behaviors.
- Deliver consistent sensory feedback for gesture-based actions.
- Ensure gestures and animations are intuitive and predictable, aligned with real-world applications.

Table 7 shows quantitative data on the results based on themes.

TABLE 7

Key Findings Based on the Themes

Theme	User Preference
Embodied Interaction	85% prefer natural gestures
Context-Aware Behavior	70% want an adaptable UI

Accessibility & Customization	62% want customization options
Modular & Minimal UI	70% want unobtrusive, floating elements
Gesture/Feedback Consistency	75% want a global interaction grammar

4.1.3 Discussion

All the participants shared their imaginative ideas on how XR should feel and behave. In Focus Group 1, one of the ideas was having a cartoon avatar functioning as a conversational agent during onboarding, instead of having a text-based introduction. The participants also suggested the importance of being able to customize the environment and letting users change the environment settings like language, interface color schemes, or assistant voice, during the onboarding, to create a more adaptable, emotionally attuned environment. Also, the ability to use the combination of gesture and button-based controls was discussed to create a bridge between the real world and the XR environment.

The second focus group really emphasized modularity. They wanted universal gestures like "undo" or "help" that work the same across different platforms. This would make learning easier and give users more control. Their ideas showed how important it is to have adaptable systems that can adjust to each person's preferences and how they use the system.

Focus Group 3 proposed more embodied interactions compared to other groups, such as mid-air gestures like finger-rotating for volume control or self-hugging to access support or feedback. Environmental customization is also discussed, similar to group 1, and one of the examples was being able to configure ambient sounds.

4.1.4 Co-Designed Interaction Components

The co-design workshops allowed participants to take part not just by giving ideas, but by actively helping design the XR components. During each session, they sketched ideas and worked together to go through possible interaction flows. This led to the creation of several clear and useful component concepts. Based on the feedback from the focus groups and workshops, the following Table 8 shows components that are conceptualized for prototyping and user testing.

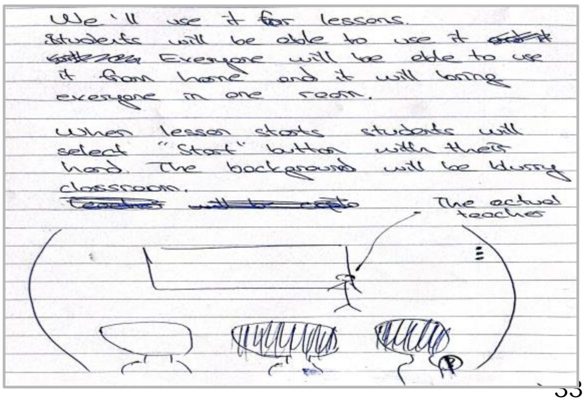
TABLE 8

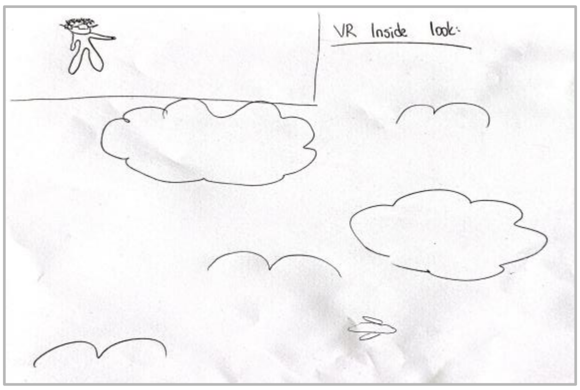
Conceptualized Components Based on Co-design Workshops

Category	**Functionality**	**Derived From**
Menu & Navigation	Circular tool menu that activates on gesture, rotates for option selection	FG3 (concept sketches), FG2 (minimal UI)

	Menu that appears when tapping or looking at hip/wrist/shoulder	FG3 (P1: “tap pocket”), FG1 (wrist gesture)
	Menus summoned via voice command that shows where you are	FG2, FG1 (preferred for accessibility/fatigue)
	Menu that appears at startup to let user select language, theme, and environment	FG1, FG3 (personalization needs)
Context & Feedback	Info panel that appears after brief focus (e.g., 2–3 sec gaze dwell)	FG3 (P3: dwell = info), FG2 (gaze selection)
	Confirms actions via gesture + proximity rather than clicking	FG3 (gesture-based confirmation ideas)
Personal Dashboards	Floating dashboard anchored to the user's wrist for quick access	FG2 (dashboard on wrist), FG1
	Toggle between gesture, voice, gaze, and controller-based input	FG2, FG1 (accessibility and flexibility)
Shortcut Interactions	Custom gestures (e.g., hug for help, rotate finger for volume) trigger quick actions	FG3 (hug = help, rotate finger = music), FG2
	Rotating gesture in mid-air to control volume or media	FG3 (music interaction: finger rotation, “touching the sky”)
Onboarding & Learning	Universal or required gestures, voice, and contextual instructions introduced by an avatar	FG1, FG3 (onboarding critique & preference for agents)

These components were designed with users and represent possible DLS elements that go beyond static UI components, which highlights the importance of the components that align with spatial awareness, bodily cues, and contextual responsiveness.





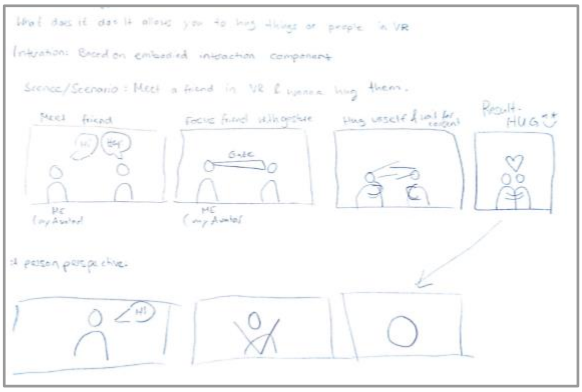


Fig.8,9,10. Co-design sketches from each group

4.1.5 Summary

These results show that it's important to design XR components that are consistent, flexible, and responsive to context. DLS should focus not only on how things look, but also on how they work and adapt. It should include interaction patterns as reusable building blocks. This user-focused approach helps create XR environments that are not only visually unified but also practical, accessible, and easy to scale across different apps and platforms.

4.2 Card Sorting Activity with Philips Designers & Developers

In parallel with the co-design workshops, a second participatory activity was conducted with internal Philips stakeholders to understand how existing components from the DLS are intuitively perceived for XR. This session, structured as a hybrid card sorting activity, which means that the participants were provided with pre-determined categories but also were allowed to create their own. With this activity it is aimed to explore how stakeholders interpret the adaptability of web-based UI elements within immersive environments, it is important to keep in mind that most participants in the card sorting activity were aware that the components originated from Philips' DLS, which allowed them to reflect intentionally on how such elements could or should translate into XR. Their insights were especially valuable in revealing how experienced designers and developers intuitively map familiar components to new interaction contexts. However, a few participants were not explicitly aware that these components were originally designed for web interfaces. As a result, their categorizations were based on first impressions and functional expectations, offering a more intuitive and unbiased perspective. This dual approach elaborated the results, offering valuable interpretations of the system's cross-platform potential.

4.2.1 Methodology

4.2.1.1 Participants & Recruitment

In total, 7 Philips employees, including designers and developers, participated in the activity. Participants were selected from different teams and spanned a wide range of ages, professional roles, and nationalities, reflecting the international environment of Philips. The activity was conducted remotely via an online collaborative board, which was created on Miro. A card sorting template was used and customized using the DLS elements; pictures were also added to increase the clarity so that users can understand the function of each component, if they are not familiar with it.

4.2.1.2 Activity Structure

The card sorting activity was designed in a hybrid manner, including both closed and open categories to foster intuitive decision-making and creativity. The activity began with three predetermined categories, which were introduced as follows.

XR – Seen as suitable for XR environments
Web – Seen as suitable for 2D interfaces
Unsure – Ambiguous or undecidable elements

Participants were also allowed to create their own categories if the existing ones didn't fit their ideas. Two empty boxes were placed next to the given categories so they could add new ones. This was explained to them at the start of the activity in the introduction box which can be seen in Fig.11.

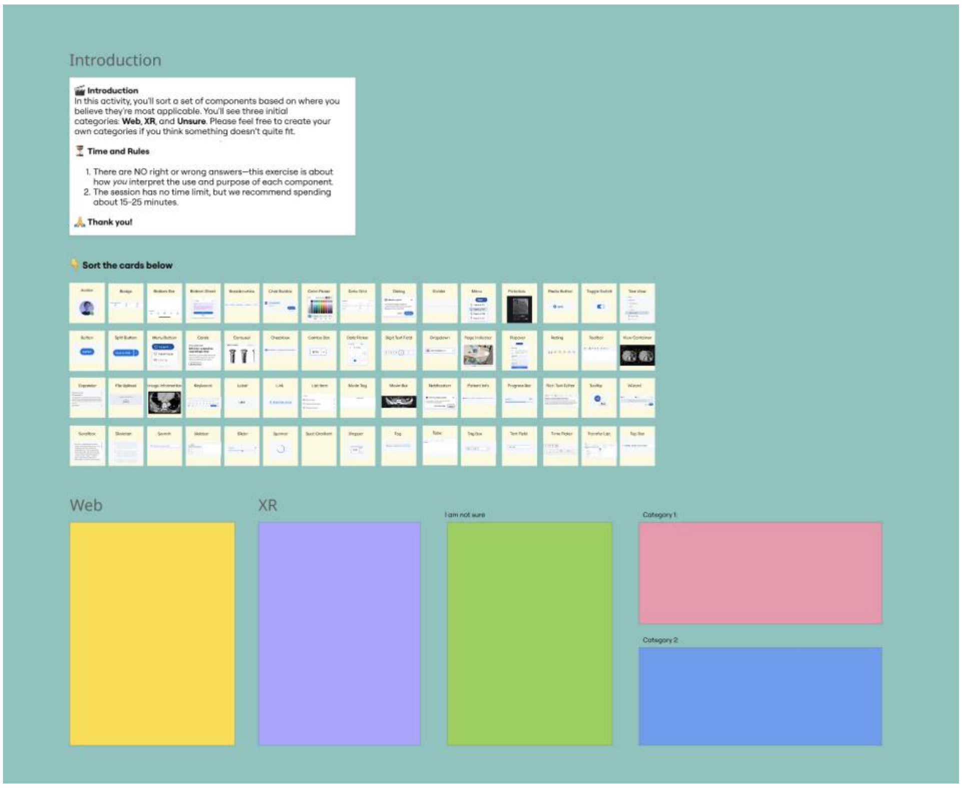


Fig.11. The card sorting board on Miro

4.2.2 Results of Card Sorting Activity

The activity was completed by 7 participants; the participants were from both the design and development teams. Each participant categorized 60 components based on their perceived suitability for Web, XR, or Unsure.

Notably, all participants independently introduced an additional category labeled "Both" with slight variations in naming, indicating a shared understanding that some components are equally applicable to both Web and

XR interfaces. This reflects a strong perception of hybrid adaptability within current design systems.

4.2.2.1 Distribution of Components Across Categories

The analysis included 420 total component placements (60 components × 7 participants). Of these, 186 placements were marked as "Both" and 74 as "XR-only." This means that over 61.9% of the components were considered suitable for XR environments, either exclusively or alongside Web usage.

The Components consistently categorized under both Web and XR are shown in Fig.12.

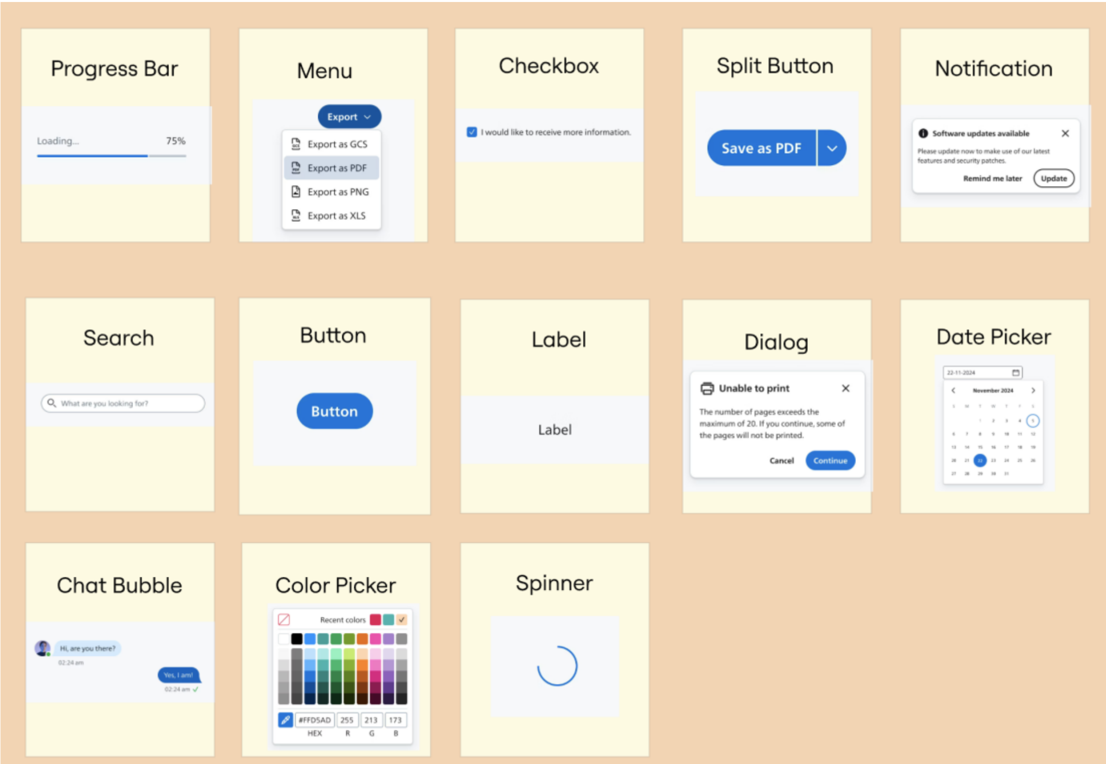


Fig.12. Components consistently categorized under web and XR

Below you can find the frequency table of components based on how many participants categorized them under both Web and XR.

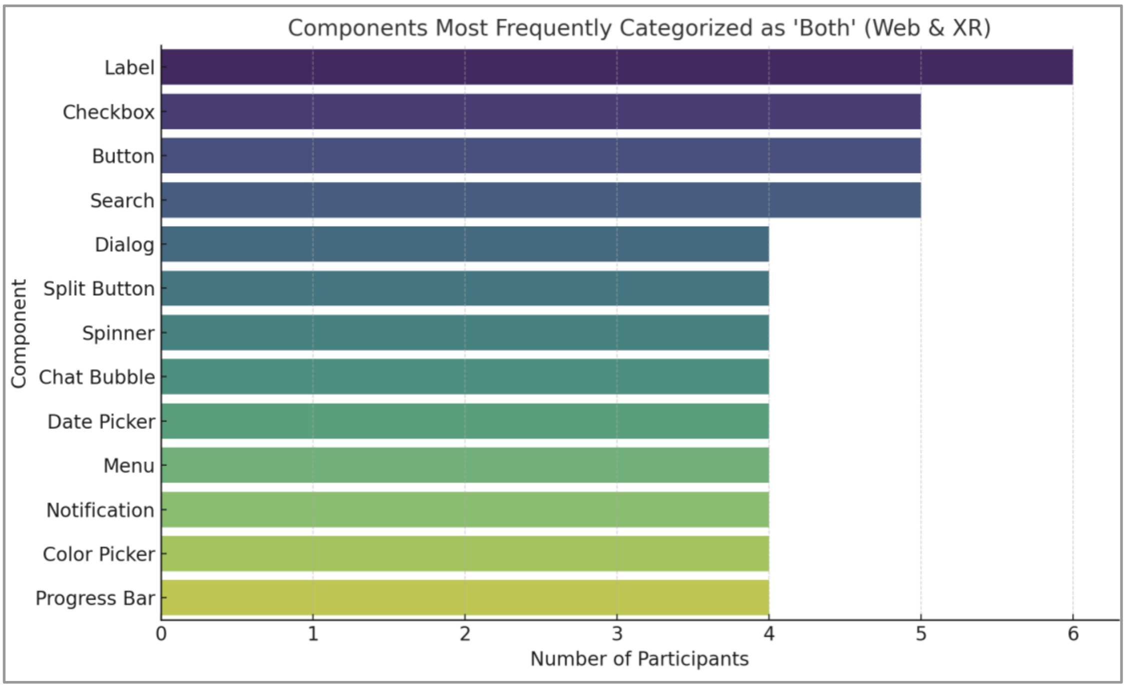


Fig.13. Components most frequently categorized as 'both'

Components most often categorized under the Web-only category included are shown in Fig.14.

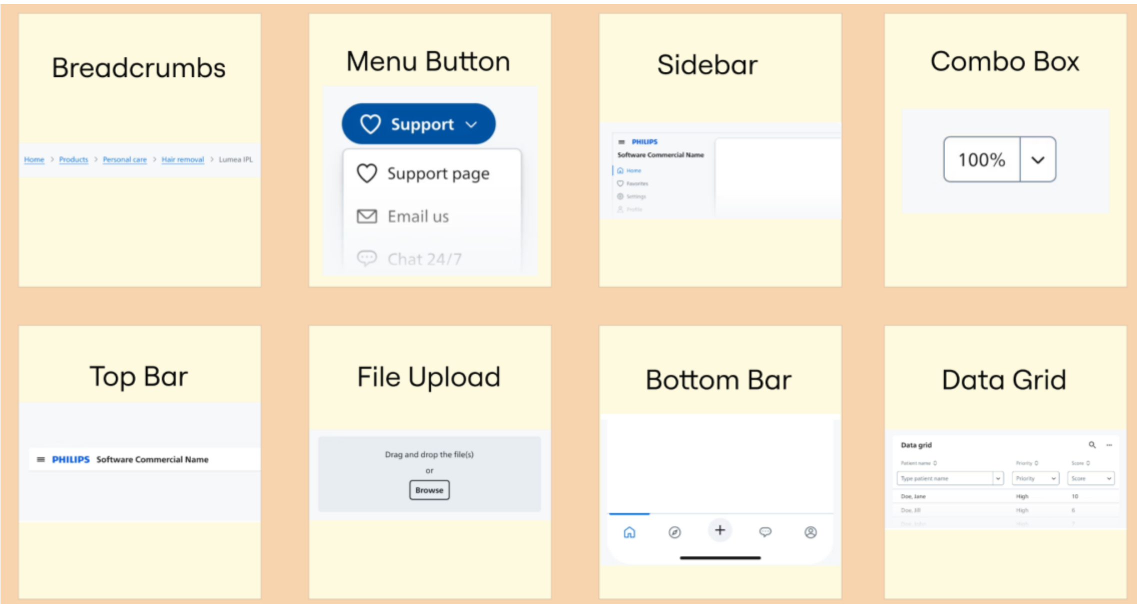


Fig.14. Components categorized under Web-only

Following, the frequency of components is given based on how many participants categorized them under Web only:

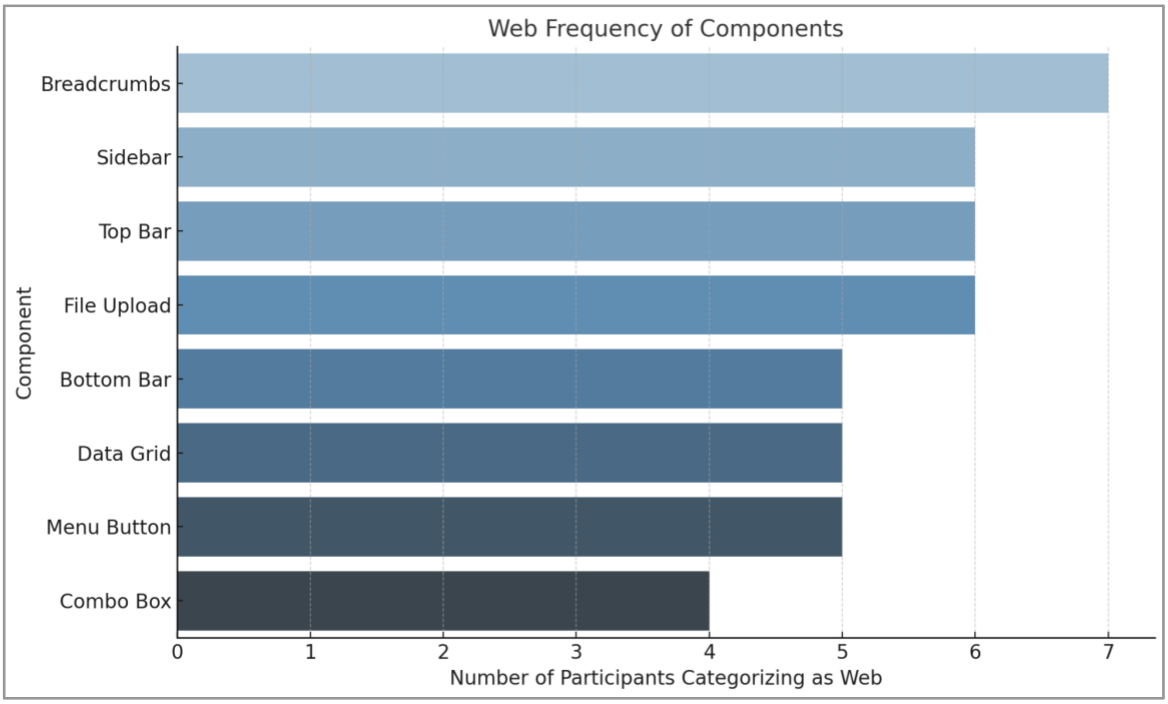


Fig.15. Web frequency of components

XR-only components included: Avatar, Carousel, View Container, Image Information, Keyboard, Skeleton, and Toggle Switch, as can be seen in Fig.16. Only the ones that are categorized under XR-only by 3 or more participants are shown below (see Fig.17).

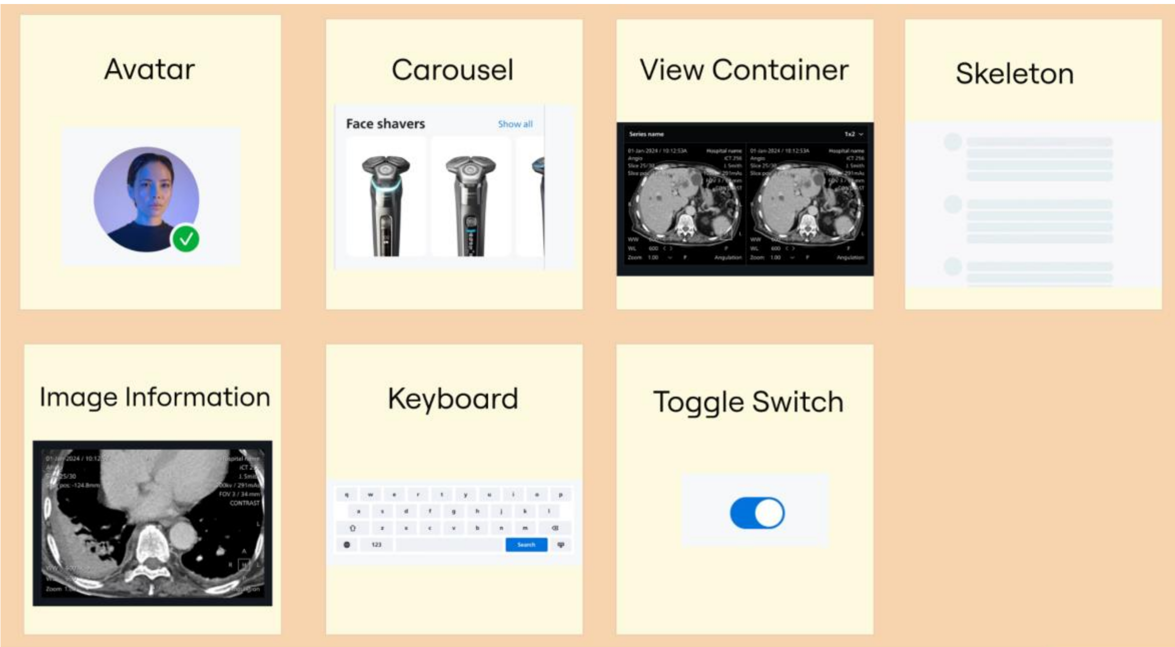


Fig.16. Components categorized under XR-only

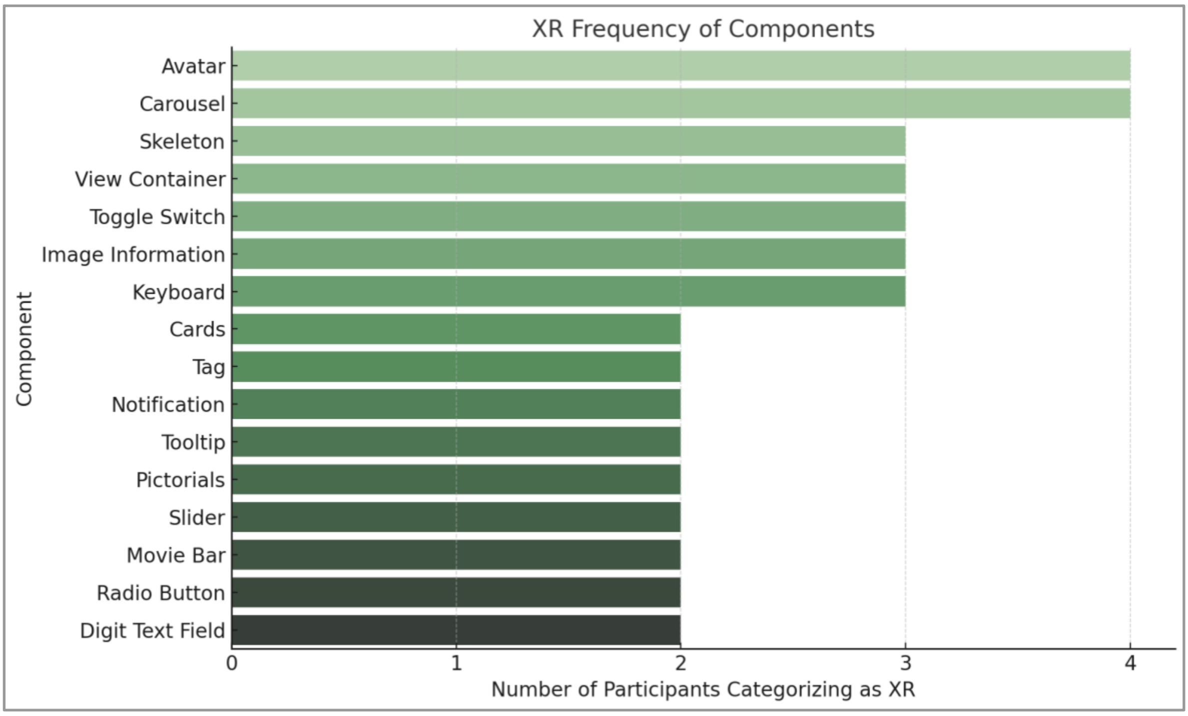


Fig.17. XR frequency of components

4.2.2.2 Components with Diverging Interpretations

Some components were placed in different categories by different people. These include **Popover**, **Wizard**, and **Stepper**. They were often put in "Both" or "Unsure," which shows that their use is not very clear or depends a lot on the situation.

4.2.2.3 Use of the "Unsure" Category

Every participant used the "Unsure" category at least once. Common components placed here include **Spot Gradient**, **Mode Tag**, **Bottom Sheet**, and **Rating**. These may be less familiar in daily design work or seen as new or mixed-use elements.

Some of the data from participants are shown below in Figs. 18,19, and 20.

Fig.18. Card sorting board of participant 2

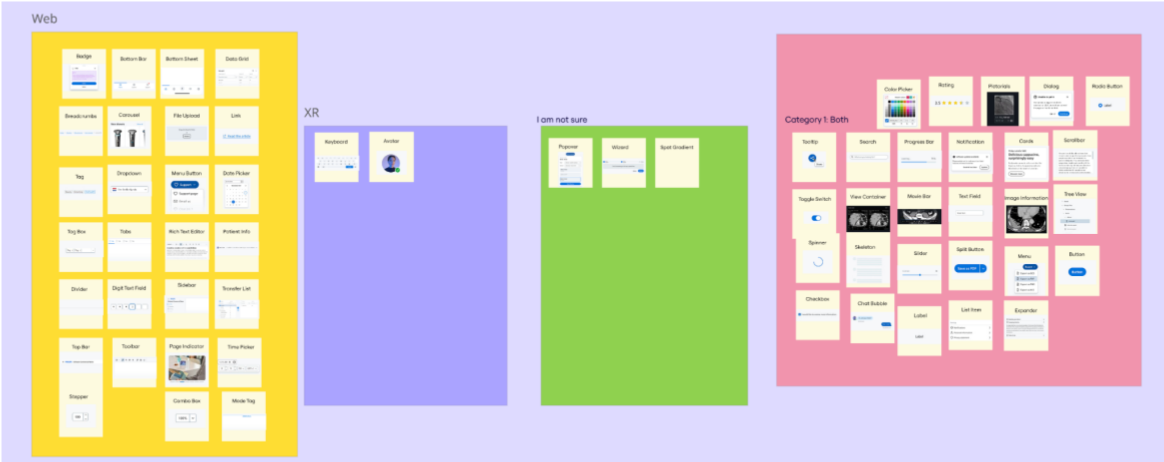


Fig.19. Card Sorting board of participant 5

Fig.20. Card Sorting board of participant 7

4.2.3 Discussion of Card Sorting Activity Results

This activity provided valuable insights on how the Philips stakeholders perceive the adaptability of DLS components across platforms. One of the most interesting findings was the creation of the "Both" category by all participants, which shows that the majority of participants agreed that most of the components are no longer related to only Web or XR but exist across both, adaptable through appropriate touches.

Out of 406 total component placements by 7 participants, 250 placements were considered XR-relevant, either categorized as XR-only (53) or placed in the self-defined "Both" category (197). This means that 61.6% of all placements indicate some level of perceived XR compatibility. Rather than requiring entirely new paradigms, these results suggest that most of the existing DLS can be translated into XR through gesture, gaze, or spatial adaptations.

Participants consistently categorized foundational UI elements like Button, Label, Search Bar, and Notification in the "Both" category. This shows that participants mostly agreed that these components work well on both Web and XR. They saw them as flexible and easy to adapt, aligning with interview results, with the main difference being how users interact with them, like tapping on a screen for Web or using gestures in XR.

In contrast, components like **Breadcrumbs**, **Sidebar**, **Top Bar**, and **Data Grid** were regularly categorized under Web-only. These items depend a lot on 2D layouts or mouse-based actions, which don't work well in XR. XR uses space and movement differently, so these web-style elements don't fit naturally in that environment.

A small set of components, such as **Mode Tag**, **Bottom Sheet**, and **Spot Gradient,** frequently appeared in the "Unsure" category. This suggests ambiguity in either their definition or usage in XR contexts. This means Philips' DLS should give better explanations and examples for these components to help people understand when and how to use them.

On the other hand, components like **Avatar, Keyboard, and Carousel** were usually placed in the XR-only category. Participants saw these as tied to spatial or embodied interaction, which makes them hard to use in regular web interfaces. This shows the importance of including XR-specific components in Philips' Design Language System, since they support interactions that only make sense in immersive environments.

Focusing on both hybrid components and XR-only elements will help make the DLS more flexible and ready for the future. However, it is important to keep in mind that this study was conducted with a relatively small participant sample. While the results offer valuable insights, they may not fully represent the broader range of views across all design and development teams. Future studies with more diverse participants could strengthen and validate these findings further.

4.2.4 Summary

This study shows that many interface components can work in both Web and XR if they are adapted carefully. About 61.6% of the placements were seen as at least partly suitable for XR, which suggests that current design patterns don't always need to be completely changed. The fact that all participants created a "Both" category shows that designers expect more flexible components that work across different platforms.

To keep up with this shift, Philips' DLS should support both flexible components and create space for new XR-native elements. These XR-specific parts need their clear guidance, since they rely on spatial or body-based interaction.

5 XR Components to Design

This section shows four XR interaction components designed for prototyping. They are based on common ideas and feedback from the focus group and co-design sessions. The components address user needs such as keeping the interface simple, providing contextual feedback, and supporting body-centered interaction. The low-fidelity prototypes created in Figma will be presented in Section 6.2.

5.1 Navigation Assist Panel

Category: Navigation

Functionality: Provides an always-accessible orientation panel with a location marker and directional cues. Can be anchored to the user's wrist or remain environment-fixed. Activates upon voice request.

User Insight: Users in the focus groups often mentioned feeling lost or disoriented in immersive environments. FG1 participants said, "I couldn't tell where to go next," and FG3 participants requested features like directional lights or floating maps. Stress during navigation, especially in tasks, was also highlighted by FG2. The idea of the panel was well-liked because it helps users stay oriented without getting in the way. Some suggested placing it on the wrist for easy access or in a corner to keep it out of the way. The design aims to give helpful guidance while keeping the interface simple.

5.2 Radial Menu

Category: Menu Access

Functionality: Activated by palm-open gesture or voice. The radial menu appears:

- Centered ~1m in front of the user at waist or chest level

OR

- Around the hand or controller ray intersection point (if gesture-activated)

Dismissible or collapsible after use. Users select by gazing and pinching at icons. The menu lets users quickly do things like change the theme, turn help tips on or off, or open the settings.

User Insight: This idea came from FG3 as a simple and easy way to access tools without filling up the screen. FG3 participants sketched radial menus during co-design sessions and said, "I want the tools to surround me, not stack up." FG2 emphasized reducing arm movement and avoiding layered UI. This component answers their feedback by supporting personalization and reducing visual clutter, while still letting users switch between tools quickly. The round, circular layout feels familiar like real-world tool use and was well liked in early design sketches.

5.3 Gaze-Aware Tooltip

Category: Context & Feedback

Functionality: Displays label, hint, or description. Tooltip appears after the user fixates on an object for 2–3 seconds on the object with the "Tap for details" label. Tapping the button opens a brief explanation or extended tooltip content. Fades away when the user looks away.

User Insight: This concept was suggested more than once by FG3 participants because it helps reduce physical effort and makes the system easier to use. FG2 also liked that gaze-based interactions felt natural and were helpful for people who may be tired or have limited movement. Many participants from both groups said they needed hands-free systems, especially in places like hospitals where touching things isn't always possible. They wanted tooltips that only show up when needed, instead of filling the screen. This design is good for new users because it gives simple feedback at the right time without needing extra actions.

5.4 Gesture-Onboarding Overlay

Category: Onboarding / Gesture Guidance

Functionality: Activates during first-time use to show the existing gestures, or after repeated failed interactions, or upon user request by clicking the help button.

Displays gesture hints (e.g., pinch, swipe, grab). Users receive contextual instruction passively, with optional voice guidance. Includes an optional "Tap for Demo" button that appears under the gesture hints, offering a quick in-context walkthrough or demo using the shown gestures.

User Insight: This component is based on feedback from FG2 and FG3 about problems with onboarding. FG2 participants said, "Sometimes I don't know what I'm supposed to do. I'd love it if it just showed me how." FG3 emphasized the need for passive help that doesn't interrupt tasks. Many participants were hesitant or confused when learning gestures; this overlay meets that need by giving guidance in the moment, without overwhelming the user. It works for people with different experience levels and is especially useful in technical or medical settings, where learning gestures quickly and smoothly is important.

5.5 Usability Testing: Future Evaluation Plan

As part of the further development of the proposed XR components, structured usability testing is planned to assess their effectiveness, usability, and alignment with user needs identified during earlier research phases. The testing process will combine both qualitative and quantitative methods to gather well-rounded insights.

A set of targeted questions has been prepared to guide post-task interviews and think-aloud protocols. These questions are designed to focus on key interaction aspects such as clarity, comfort, responsiveness, and perceived usefulness while avoiding overlap. They aim to support early feedback collection once interactive prototypes are available.

Planned questions include:

Navigation Assist Panel

- How did you understand your current location while using the panel?
- What did you think about where the panel was placed?
- What was your experience with the directional cues?
- In what ways did the panel affect your sense of orientation?

Radial Menu

- How did you activate the radial menu?
- What was your impression of the circular layout?
- Can you describe your experience selecting an option?
- How did you feel about the spacing between icons?
- What was it like to perform the gesture needed for the menu?

Gaze-Aware Tooltip

- How did you feel about the timing when the tooltips appeared?
- How would you describe the readability of the tooltips?
- What did you notice about how long tooltips stayed visible?
- Were there any moments when tooltips felt distracting?

Gesture-Onboarding Overlay

- What kind of help did the system provide while you were interacting?
- How did you respond to the gesture hints shown during your task?
- What effect did the visual guidance have on your ability to complete the task?
- How did the onboarding visuals affect your overall experience in the environment?

To complement observational data in future stages, the System Usability Scale (SUS) can be used to provide standardized usability metrics, offering a benchmark for ease of use and user satisfaction. Prototypes developed in Unity or WebXR will be tested through realistic, task-based scenarios. Sessions will involve observing user interaction, collecting feedback through guided prompts, and conducting short interviews.

This planned testing phase is not included within the scope of this thesis but is intended for a future evaluation cycle to validate the design decisions, identify friction points, and inform necessary refinements before integration into a broader XR DLS.

6 XR UI Guide

This guide is intended for UX designers, design system leads, and XR interface developers working on immersive simulations. It outlines adapted design principles and interaction components derived from the original Philips DLS, informed by card sorting, and also the new components informed by co-design workshops. It acts as a bridge between traditional 2D design systems and spatial XR interfaces, supporting consistent, intuitive, and accessible user experiences across immersive platforms.

6.1 Transformed DLS Elements for XR

In this section, the existing 2D DLS elements that are categorized by participants under the Web, XR, or Both categories will be tackled and adapted to integrate them into XR environments.

6.1.1 Adaptable Web Components

The following components were most often placed in the “Both” category during the card sorting activity by participants, meaning they are seen as useful in both Web and XR. This shows a shared view that, with some small changes, these elements can work well across screens and spatial environments. The design tips below are based on Philips’s DLS and also inspired by other platforms (see Section 2.6) to help guide early testing at Philips [21,22,23,24]. These references [21–24] were used not to copy exact heuristics, but to extract spatial layout principles, input recommendations, and accessibility guidelines relevant to XR interaction design.

Label

Adaptation: Anchored within spatial UI panels; maintains a flat 2D appearance to maximize legibility.
Typography: Use high-contrast, semibold fonts.
Size & Placement: Font size ranges from ~14–18pt (or ~60pt depending on depth), positioned at eye level, typically ~1.5–1.75 m from the user [21, 24].
Feedback: Avoid subtle fades or motion-based text that hinder readability in spatial contexts [21].

Checkbox

Adaptation: Rendered as volumetric toggles or 3D tick boxes.
Interaction: Designed for raycast or direct hand input. Ensure collider zones are ≥ ~3.2 cm [23, 24].
Feedback: Use color changes or brief audio cues to signal state. Maintain ≥ 8dp spacing to prevent errors [22].

Button

Adaptation: Appears as floating or surface-mounted spatial control.
Size & Placement: Minimum diameter ~3.2 cm, with ~60pt spacing; optimal distance is 0.5–2 m [21, 23, 24].
Interaction: Supports gaze dwell, raycast, or hand pinch.**Feedback:** Include hover states (glow/pulse) and subtle motion/lighting for active states [21, 23].

Search

Adaptation: Presented as a floating panel or modal field.
Input Methods: Includes virtual keyboard, voice, or gesture typing [21, 24].
Placement: Centered in the user's field of view for focus-intensive tasks.

Dialog

Adaptation: Displayed as world-locked overlays or positioned near the content it relates to. Avoid head-locked placement.
Size & Placement: Responsive layout, max width ~70% FOV, positioned ~1–1.75 m away [21, 24].
Feedback: Use soft fades, minimal transitions, and avoid large-scale motion [21].

Split Button

Adaptation: Combines a main action and secondary trigger into one control.
Labeling: Icons or text with ~60pt spacing and ~8dp padding [21, 22].
Interaction: Submenus accessed by gaze or pinch. Test for occlusion and fatigue in long tasks [23, 24].

Spinner

Adaptation: Circular progress indicators in spatial context.
Motion: Favor eased rotation and avoid flickering [21, 23].
Feedback: May include subtle sound or haptic feedback on completion.

Chat Bubble

Adaptation: Curved floating panels tied to avatars, agents, or notifications.
Use Case: Real-time messaging or system feedback.
Placement: Near the speaking avatar or anchored to content; avoid visual clutter.

Date Picker

Adaptation: Spatial dial, scroll column, or calendar carousel.
Interaction: Supports pinch, scroll, or gaze input. Use snapping and smooth transitions [23].
Placement: Anchored near the related field, avoiding peripheral or awkward viewing zones [21].

Menu

Adaptation: Can appear as radial wheels, carousels, or wrist UIs.
Interaction: Gaze or low-effort gestures like ray select or pinch.
Placement: Positioned ~1.5 m from the user and aligned with gaze [21, 24].

Notification

Adaptation: Displayed as anchored alerts or floating toasts.
Interaction: Auto-dismiss or gesture to close.
Placement: Placed ~1.5 m from the user, fading in/out gently [21, 24].

Color Picker

Adaptation: Carousel or radial selector in 3D space.
Interaction: Ray hover, pinch, or swipe to preview/apply color [23].
Placement: Near the object being edited or the user's dominant hand. Ensure contrast and legibility at ~1–2 m [23].

Progress Bar

Adaptation: Horizontal or radial loading indicator.
Placement: Near the task or centrally anchored if modal.
Feedback: Use progressive fill, soft scaling, or color shifts. Avoid full-screen takeovers unless necessary [21, 24].

6.1.2 XR-only Components Identified by Participants

The following components were categorized under XR by at least three participants in the card sorting activity. These elements reflect a shared view that their interaction style, presentation, or purpose is inherently tied to spatial and embodied contexts. The adaptation notes below establish initial design benchmarks, grounded in the Philips DLS and shaped by insights distilled from platform practices discussed in Section 2.6.

Avatar

Use: Represents digital assistants, onboarding agents, or conversational guides.
Placement: World-anchored, positioned ~1.5 m away at eye level [21, 24].
Animation: Subtle idle animations and responsive eye contact foster engagement. Avoid sudden movements to maintain user comfort [21].

Carousel

Use: Presents scrollable media, tools, or object sets in 3D space.
Interaction: Supports pinch, swipe, or ray-based navigation. Infinite looping should be paired with clear edge feedback [23].
Placement: Anchored at ~1–2 m with a curved layout that maintains visibility within a ~40–60° field of view [21, 23].

Skeleton

Use: Placeholder for content during loading or progressive rendering.
Visual Style: Semi-transparent blocks or blurred layout hints [24].
Animation: Use soft fade-ins or pulsing transitions. Avoid sudden movements [21, 22].

View Container

Use: Groups spatial UI blocks such as panels or toolbars, similar to card-based layouts.
Function: Enables dynamic scaling and visibility toggles.
Placement: World-locked and depth-consistent (~1.5 m) to support spatial stability [24].

Toggle Switch

Use: Binary state control using sliders or knobs.
Interaction: Ray, gaze + dwell, or direct poke. Colliders should be ≥ ~3.2 cm for reliable selection [23, 24].
Feedback: Use color transitions and (if available) haptic cues. Prefer tactile metaphors over flat UI [22].

Image Information

Use: Displays metadata for 3D objects or scenes.
Placement: Anchored adjacent to the object and within 1–1.75 m viewing range [21, 24].
Format: Include text and thumbnails; allow expansion via gaze to preserve clarity [22].

Keyboard

Use: Virtual text input using gaze, hand, or voice [22].
Placement: Split keyboard placed at ~0.7–1.2 m for ergonomic reach [24].
Input Modes: Gaze typing, poke input, and optional voice.

These components work well with body movement, gaze, or gesture-based interactions. The way participants grouped them shows they share a similar understanding of what makes a good XR experience, which is different from regular screen-based design. Also, these components match the ideas used in top XR platforms which suggests that Philips' design elements can be used successfully across both Web and XR systems.

6.2 New XR Interaction Components

To support the adapted DLS elements, four XR-native components were prototyped using Figma based on co-design sessions and feedback from user tests. The layout and component structure of the prototypes were adapted from an open-source Figma template [29], and then modified to align with Philips' DLS and user feedback. These components aim to solve common usability issues by making navigation easier, simplifying tool access, providing contextual help, and keeping the interface lightweight and intuitive.

During the co-design workshops, participants highlighted the need for clear cues, simple interaction, and reduced complexity. They said they preferred elements that were easy to notice and understand, what they called *"transparent"* or *"visible by design."* This directly influenced the design of the Gaze-Aware Tooltip, which gives help automatically when the user looks at an object for a few seconds, and the Radial Menu, which shows a circular menu around the hand or pointer when triggered by a pinch. Both examples align with what Vieira et al. [3] describe as *perceptible affordances*, design elements that are visible and easy to discover. For more advanced or hidden features, users need guidance, which is exactly what the Gesture-Onboarding Overlay provides by showing animated hints and instructions during first-time use or after repeated failed attempts.

The *placement* of these interactive elements was also important. Vieira et al. note that XR systems work best when affordances are spread across the body, tools, and environment [3]. This is reflected in how the Radial Menu appears near the hand or raycast point, making it feel natural and integrated into the user's movements. The Navigation Assist Panel, on the other hand, appears either fixed in the user's field of view or as a small panel attached to the top of the wrist, which users can access by raising their arm, helping them stay oriented without cluttering their view.

These components also support key ideas from the Cognitive Affective Model of Immersive Learning [6], which highlights the importance of *interest, confidence, embodiment,* and *low cognitive load* in XR. The Gesture-Onboarding Overlay helps build confidence by guiding users in the moment. The Navigation Assist Panel makes it easier to understand spatial environments. The Gaze-Aware Tooltip prevents information overload by only appearing when needed. And the Radial Menu keeps users engaged by using familiar gestures like pinching and hand tilting to select tools.

By combining participatory insights, cognitive learning models, and affordance design principles, these four components help define a more usable and thoughtful approach to XR interaction.

Component: Navigation Assist Panel

Category: Spatial Orientation / Environmental Guidance

Purpose: Guides users through room-based transitions in immersive simulations, showing exits, destinations, or re-orienting cues.

Visual Preview: Flat panel with arrows, labels, exit, and a return to start button.

Placement: World-locked 1m in front, head height (1.4–1.6m from ground). Optionally, the panel can be anchored to the user's non-dominant wrist and activated through a "raise-to-view" interaction. The wrist version provides mobile, glanceable access to spatial guidance without blocking the user's main field of view.

Trigger: Voice: "Where am I?" or auto-trigger on room entry or task completion.

Interaction: Clickable via hand poke or raycast. Voice narration for wayfinding is supported. Users may select destination markers, see directional arrows, or use a "return to start" function.

States:

Default: Hidden (icon hint only)
Active: Assist panel expands, voice or text guidance

Accessibility Notes: Fixed elevation to match ergonomic eye-level range, logical directional cues, high-contrast visuals, and multi-language label support. Wrist-anchored version respects handedness and includes fade-out when not in view.

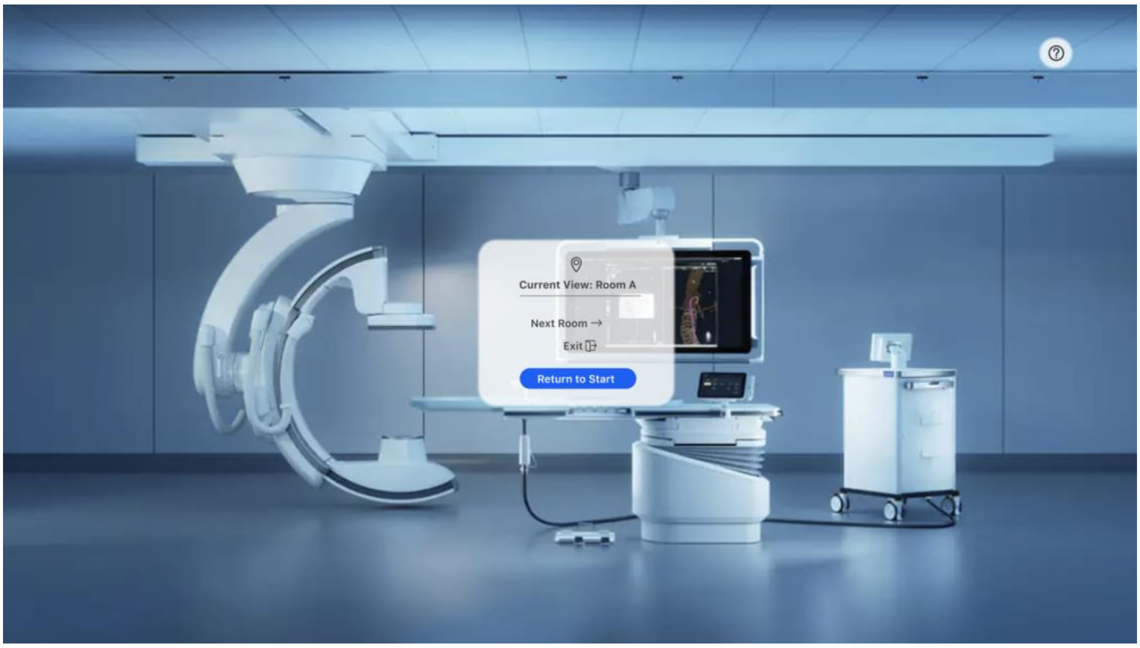


Fig.21. Figma prototype of navigation assist panel

Component: Radial Menu

Category: Tool Selection / Gesture-Based Menu

Purpose: Enables natural, low-effort access to tools using a radial menu that wraps around the hand or pointer location.

Visual Preview: 360° circular menu expanding from the palm or ray intersection point.

Placement: World-relative to hand or pointer location, ~0.8–1m distance.

Trigger: Gesture: Pinch + hold, voice: “Show Tools”

Interaction: Icons appear in an arc or full circle around the user’s hand or ray. Tools can be selected using gestures like pinching or tapping. The menu collapses automatically on selection or release.

States:

Default: Hidden (icon hint only)
Active: Expanded icon ring
Disabled: Transparent center

Accessibility Notes: Icons ≥ 3cm in diameter for reliable selection, spaced to minimize overlap. Voice labels on hover. Designed to avoid vertical occlusion and limit wrist fatigue.

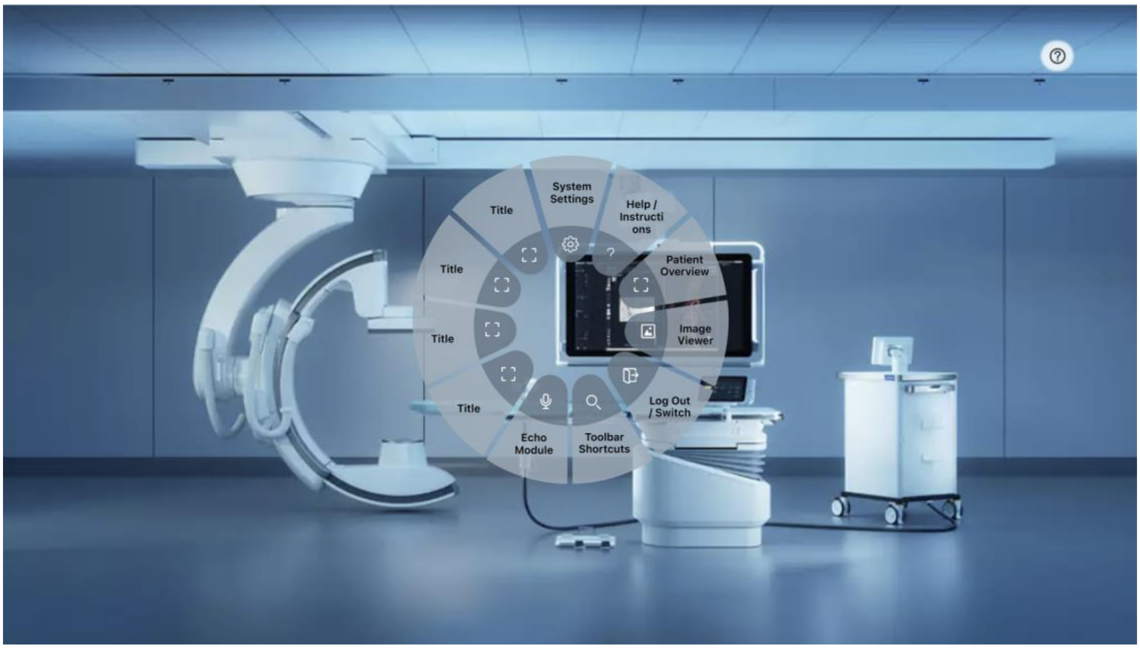


Fig.22. Figma prototype of radial menu

Component: Gaze-Aware Tooltip

Category: Contextual Feedback

Purpose: Delivers non-intrusive help or information after gaze focus, reducing cognitive overhead during tasks.

Visual Preview: Floating button above the target object or button.

Placement: Object-relative; appears under the interactive element.

Trigger: Gaze dwell (2–3 seconds)

Interaction: Appears automatically when the user maintains gaze; fades out when gaze shifts or after a fixed duration. When the button is tapped, the pop-up disappears by pressing the close button.

States:

Default: Hidden
Active: Fades in and remains for 6 seconds
Inactive: Smooth fade-out

Accessibility Notes: Contrast and text size are adjustable; the tooltip is hidden during rapid movement. Optional voice-over for visually impaired users. Placement avoids occluding key UI elements.

Fig.23. Figma prototype of gaze-aware tooltip

Fig.24. Figma prototype of gaze aware tooltip when the button is tapped

Component: Gesture-Onboarding Overlay

Category: Onboarding / Gesture Guidance

Purpose: Provides real-time gesture assistance based on context, supporting onboarding and reducing learning barriers.

Visual Preview: Gesture hints (e.g., pinch, swipe) on a flat panel floating near the relevant object or in front of the user during onboarding.

Placement: Adaptive appears near the UI element, or in the center of view, based on task context.

Trigger: Auto-triggered during first-time use, after tapping the question mark button at the upper right corner, or via voice command: “Gesture Help.”

Interaction: Passive overlay that users can dismiss or replay. Overlay appears with subtle animation and optional audio guidance.

States:

Default: Off
Active: Visible, help icon or gesture replay option

Accessibility Notes: Supports audio narration, slowed-down animations, and persistent replay options. Fade out on completion. Available in multiple languages.

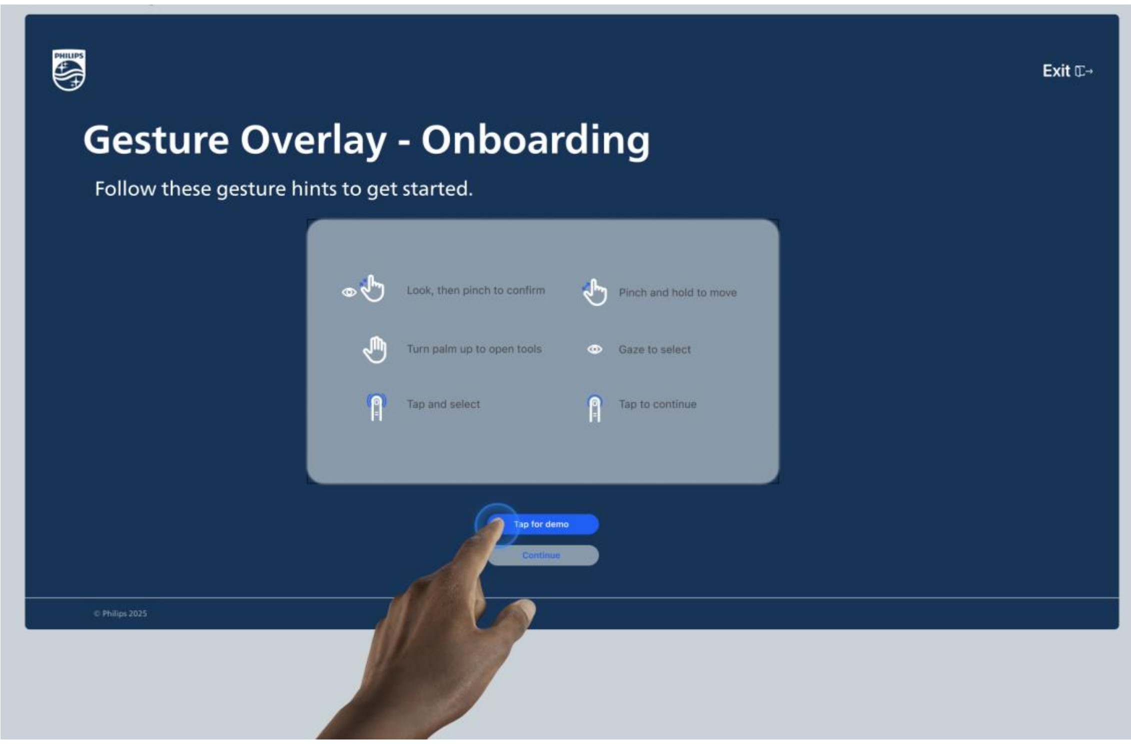


Fig.25. Figma prototype of gesture overlay on onboarding

Fig.26. Figma prototype of gesture overlay on a device

6.3 Contextual Application: In-Room Adaptation of the Navigation Assist Panel

Many XR applications at Philips are used in indoor environments like hospital rooms, labs, or training spaces. These settings often come with real-world limitations such as limited walking space, reflective surfaces, or changing light conditions that can affect how people interact with virtual content.

This section shows how the existing Navigation Assist Panel was adapted to work better in these kinds of environments, without losing its connection to Philips' DLS.

6.3.1 How It Works in Room

1. The panel gives users a quick overview of the room layout or shows arrows and signs to help them find their way.
2. It appears automatically when the user gets close to a room boundary or can be turned on with a voice command like “Show Navigation”, or “Where am I?”.
3. It floats about 1 meter in front of the user and stays at eye level (around 1.2–1.6m from the ground). It always faces the user, so it’s easy to see and doesn’t shift around.

6.3.2 What’s Included in the Design

- Clear arrows showing where to go next.
- A button to go back to the main menu or exit the space is designed with big, easy-to-tap zones and strong contrast for visibility.
- An optional mini-map that can be shown on the user’s non-dominant wrist, for quick access without blocking the main view.

6.3.3 Why This Design Works Indoor

- It follows Philips' accessibility rules like using readable fonts, big enough touch areas (at least 3.2 cm), and high contrast colors [19].
- It stays visible in bright or uneven lighting thanks to a slightly see-through background and bold text.
- It can be used in multiple ways: by hand gestures, gaze, or with sound or vibration feedback for confirmation.
- It still uses familiar DLS elements like buttons and panels, but adds smooth transitions to make the experience feel natural in 3D space.

This in-room version of the Navigation Assist Panel shows how one component can be customized for real-world settings, keeping it useful, accessible, and easy to use, even when space or lighting isn't ideal.

6.4 General XR UI Heuristics

The following guidelines provide general advice for designing spatial user interfaces that support Philips' goals for clarity, accessibility, and ease of use. These recommendations are based on Philips' DLS and distilled from common design patterns observed across the guidelines of the following platforms such as Apple VisionOS, Meta Presence Platform, Android XR, and Microsoft MRTK. The values provided serve as reference points and should always be validated within Philips-specific clinical and user environments before being incorporated into the official DLS.

Color and contrast
Colors need to work well in both virtual and real-world settings, especially in environments like hospitals with varying lighting conditions.

- Stick to the Philips DLS color palette when possible, but avoid using very dark colors (especially near-black) in AR, since they can disappear on headsets like HoloLens due to how light is added to the display.
- Use translucent background panels (for example, see-through cards) instead of solid black to help the UI blend better into real environments.
- Apply high-contrast accent colors to highlight interactive elements, following Philips' visual clarity principles.

Typography
Text in XR needs to stay readable from different distances and across different devices.

- As a general rule, use around 18pt font size for content that appears approximately 1.5 meters away [21,23].
- Use semi-bold or bold text for titles and important labels so they stand out clearly in space.
- Avoid using thin or light fonts, especially over transparent or complex backgrounds, where readability can suffer.

Layout and spacing
UI elements in XR should be spaced out to avoid clutter and allow users to focus.

- Leave at least 30 to 50 centimeters of space between floating panels to prevent overlap and keep the interface clean [24].
- Place interactive components within an ideal viewing and reaching distance, roughly between 1 and 2.5 meters from the user [24].
- Try not to overwhelm the user's field of view. Keep the number of major UI elements on-screen at the same time to 3 or fewer when possible.

Input zones
Buttons and other controls should be large and easy to reach, whether tapped, looked at, or activated by gesture.

- Design buttons and other touch targets to be at least 3 to 5 centimeters wide to make them easy to select [23,24].
- If you're using gaze-based input, set a dwell time of 2 to 3 seconds before triggering an action. Consider using visual countdown rings or similar feedback to let the user know something will happen.
- Follow Philips' accessibility rules by keeping enough space (at least 10mm) between touch targets to support users with less precise motor control [19].

Motion and feedback
XR users expect feedback not just through visuals, but also through sound, motion, and even vibration.

- Use simple visual effects like glowing, pulsing, or light scaling to show when an object is interactive or has changed state. This helps build user confidence.
- Keep animations smooth and comfortable; fade-ins and fade-outs of 300 to 500 milliseconds usually work well without being distracting [21].
- In spaces where visual feedback might be missed, add audio or haptic cues (like subtle sound effects or controller vibration) to confirm actions.

To complement the design heuristics discussed above, Table 9 summarizes key spatial UI guidelines distilled from industry benchmarks [21–24].

TABLE 9

Synthesized Design Heuristics for Philips XR Interfaces

Design Aspect	Concise Guideline
Interaction Design	Use ~56dp/60pt minimum target size, with ~8dp spacing to prevent input errors.
Typography& Legibility	Use ≥18pt fonts with high contrast; avoid placing text over moving or animated elements.
Motion & Transitions	Use smooth fades (~300–500ms); avoid sudden motion; introduce movement through guided cues.

Spatial Placement	Place key UI within ~30°–45° gaze range to reduce head and neck strain.
Accessibility& Focus States	Support hover/focus/dwell states; enable eye tracking and low-effort input for motor accessibility.
Passthrough& Environment	Avoid solid black in AR; use transparency and lighting to blend UI with the real world.
Gestural Interactions	Use indirect gestures (tap, pinch) for main tasks; reserve expressive gestures for short actions.

6.4.1 Adapting the DLS for Apple Vision Pro

Currently, most of the simulations that are worked on use Apple Vision Pro as the headset; therefore, I think it is important to take into consideration the challenges that could arise when designing for Vision Pro. Challenges with the HoloLens 2 were tackled before in the state-of-the-art section, like dark UI elements becoming invisible due to additive light displays, and this section introduces a new set of considerations rooted in the Apple Vision Pro's fundamentally different display and interaction model.

Unlike HoloLens, which projects light into the real world and struggles with low-contrast UI elements, the Apple Vision Pro uses high-resolution micro-OLED displays with full-color passthrough. This enables much richer visual fidelity and contrast, allowing the DLS to reintroduce darker color palettes, finer text rendering, and layered semi-transparent panels without sacrificing visibility. In contrast to the constrained brightness levels of HoloLens, Vision Pro's display allows for more expressive visual hierarchy and spatial depth.

Interaction paradigms also shift significantly. Where HoloLens relies on direct hand gestures (e.g., air tap, bloom) and voice input, Vision Pro emphasizes gaze-based targeting combined with subtle pinch gestures. This means DLS components designed for gesture-driven selection or anchored to the wrist may need to be repositioned or restructured to work effectively in Apple's centered, eye-driven UI model. For example, wrist-based menus popular in HoloLens may not align with Vision Pro's seated-use comfort zone and eye-centric interaction style.

Additionally, Apple's Human Interface Guidelines for visionOS encourage the use of depth cues, fluid animation, and minimal visual clutter, all of which must be integrated into DLS updates. Interface elements must adapt fluidly to the user's spatial context, whether in Shared Space (floating windows in the real world) or fully immersive scenes. Unlike HoloLens, which often anchors UI to fixed points or surfaces, Vision Pro encourages adaptive UI layers that respond dynamically to user positioning, focus, and environment lighting.

To effectively adapt Philips' DLS for Vision Pro, design rules must extend beyond visual tokens and address platform-specific behavior such as how components scale in depth, respond to gaze, or transition smoothly between contexts. Just as the HoloLens required changes to color and visibility, Vision Pro demands thoughtful reworking of both interaction logic and spatial design to align with Apple's hardware capabilities and interaction philosophy.

6.5 Reflection

This XR UI guide offers a practical and research-based approach to bringing Philips' DLS into immersive environments. The components and design choices are realistic to build, easy to use, and mindful of accessibility. They are based on what users actually need, as seen in the focus groups, and follow trends in the XR industry. Overall, this guide helps shape a DLS that not only works in XR but is ready to grow with it.

7 Challenges and Limitations of Adapting the DLS to XR

While Philips' DLS is strong for building consistent and accessible interfaces in 2D environments, adapting it for XR comes with some major challenges. These challenges affect how we design, interact, build, and ensure accessibility in immersive spaces.

7.1 Lack of Spatial Interaction Models

The current DLS, especially in Filament, is built for screens. It works well for clicking with a mouse or tapping on a touchscreen. But XR needs a different approach. In XR, users interact with the space around them using gestures, gaze, or voice. This means common components like buttons and panels need to be reconsidered. Where they appear, how far they are, and how users activate them need to make sense in 3D space.

7.2 No Standard Support for XR Inputs

Right now, the DLS doesn't support inputs that are common in XR, like raycasting (pointing with a virtual beam), gaze-based actions, or hand gestures. Designers have to guess or use different platform rules (like Meta Quest or Apple Vision Pro), which can lead to inconsistent experiences. Without clear patterns, components like radial menus or wrist panels may work differently across apps.

7.3 Limited Feedback

In 2D, the DLS uses things like hover effects, pressed states, and colors to give feedback. But in XR, users expect more, like sound, vibration, and spatial motion, to confirm their actions. Right now, the DLS doesn't offer enough support for these kinds of multimodal feedback. This can make interactions feel less clear or responsive in 3D space.

7.4 Tokens Not Designed for 3D

The DLS uses a token system with sizes and layout rules based on 2D screens. XR needs something different, tokens that work with depth, scale with real-world sizes, and adapt to lighting changes. Without this, using existing tokens in XR can lead to designs that feel wrong or don't work well in space.

7.5 Screen-Focused Accessibility

Philips' DLS is strong in accessibility for 2D. It includes good rules for contrast, text size, screen readers, and more. But XR has different needs. For example, some users can't easily reach far buttons, or may rely on gaze instead of touch. Others might need spatial audio to navigate. The DLS needs to grow to include these kinds of accessibility features for immersive environments.

7.6 No Workflow from Design to XR Development

The current DLS works well with tools like Figma and frameworks like React and Angular. But XR developers use different tools like Unity, or WebXR. Since the DLS isn't integrated with these platforms, designers often have to recreate components manually or make compromises. This slows things down and can break consistency.

7.7 Missing Vocabulary for XR Design

Right now, the DLS doesn't have a shared set of terms or visual examples for XR components. There are no official references for things like floating panels, 3D buttons, tooltips in space, or wrist-based UIs. It's also unclear what states like "hover" or "disabled" mean when there's no mouse or screen. Creating a shared XR glossary would help keep things consistent across teams.

8 Discussion

This thesis explored how Philips' DLS could be meaningfully extended into XR environments through a user-centered approach. While the process resulted in a set of practical design guidelines and four XR-native interaction components, a deeper reflection is needed to contextualize these outcomes with existing frameworks, potential applications, and research limitations.

8.1 Comparison with Existing Guidelines

Compared to the original 2D DLS, the XR adaptation introduced new principles around spatial interaction, gaze, and gestural controls. While 2D DLS components focus primarily on screen-based interactions and visual consistency, the XR version prioritizes embodied interaction, accessibility in 3D space, and responsiveness to user attention and movement.

This divergence is also evident when comparing with XR-specific guidelines from platforms like Apple visionOS, Microsoft's MRTK, and Meta Horizon OS. For example, Apple's design principles encourage indirect gesture use and anchored UI panels in Shared Space mode, while Philips' XR Guide introduces a radial gesture menu that supports more expressive gestures. If a team were to apply Philips' XR Guide directly to a visionOS product, conflicts might arise, for example, around control placement, gesture fatigue, or device-specific privacy constraints (like eye-tracking limitations on some devices). These differences suggest that any cross-platform design system must include adaptation layers to respect hardware-specific constraints while maintaining core interaction logic.

8.2 Potential Impact and Application Domains

The XR Guide developed in this project could have potential benefits across several domains identified in related work. For example, prior research has shown that XR can support learning environments [6], inquiry-based pedagogy [7], and embodied learning [8]. The components designed, such as the Gesture-Onboarding Overlay and Navigation Assist Panel, could help reduce cognitive load and improve comprehension in such contexts by offering interactive scaffolding.

In healthcare, prior studies have demonstrated the value of XR for reducing anxiety, supporting onboarding, and enhancing communication in clinical settings [10], [11], [13]. Although this project did not involve direct feedback from clinicians, components like the Gaze-Aware Tooltip and the Navigation Assist Panel could be adapted for surgical training, rehabilitation, or patient orientation. Further collaboration with medical staff would be necessary to validate these use cases.

8.3 Methodological Reflection and Limitations

Throughout this study, some challenges were faced and should be acknowledged. First, the study was conducted under tight time constraints, which restricted the depth of iterative prototyping and did not allow for the development and testing of higher-fidelity XR prototypes. Although a detailed usability testing plan and question set were prepared, there was not enough time to conduct testing sessions with real users.

Secondly, while the focus group sessions provided valuable qualitative insights, the overall participant pool was relatively small, 13 individuals across three sessions, and the card sorting activity included only 7 participants. The participant groups for the focus groups were also demographically narrow, with most individuals aged between 23 and 29, and nearly all being students or early-career professionals. This limited diversity may have influenced the types of needs and expectations expressed.

A further limitation was the absence of feedback from key end users, particularly medical staff or trainees, who would ultimately interact with XR systems in clinical contexts. Their input would have been especially valuable for evaluating components aimed at healthcare use cases, such as onboarding and spatial guidance.

In addition, although the card sorting activity was shared with multiple staff members, some did not participate. This limited the variety of perspectives represented in the categorization results and may have reduced the overall balance of the findings from that phase.

These limitations suggest several directions for future work. First, participant diversity should be expanded to include a wider range of age groups, professional backgrounds, and levels of XR experience. In particular, involving domain-specific users such as healthcare workers would help validate the applicability of the components and design guidelines in real-world clinical settings. Additionally, usability testing should be conducted using medium to high-fidelity prototypes in realistic XR environments to better assess interaction behavior and accessibility. Finally, efforts should be made to encourage higher participation and engagement during collaborative activities like card sorting, ensuring more robust and representative feedback for future iterations.

Despite these constraints, the components proposed in this project are grounded in user-expressed needs and reflect recurring themes across co-design sessions. They offer a solid starting point for defining reusable XR interaction patterns within an evolving design system.

8.4 Personal and Professional Growth

I have learned a lot during this study. It helped me to explore the interaction design principles within XR while growing as a researcher, designer, and communicator. The early user research was one of the hardest parts of it, as recruiting related participants, especially designers and developers, was challenging due to the purpose of ensuring meaningful engagement. Also, it pushed me to be more adaptable, like changing the structure of the activity, for example, online for some groups, and iterate on my approach when the first test session revealed shortcomings.

The analysis of qualitative data was another time-intensive and challenging phase. Although I did not have a big sample set, translating open-ended, multilingual feedback into design implications took careful interpretation. I realized how important it is to balance creativity with structure, especially when moving from user insight to systematized design components.

The card sorting activity was useful, but it didn't have as much participation or engagement as the focus groups. This showed me that relying on just one method isn't enough, and that combining different methods gives a better picture. I was also limited by time when prototyping; I would have liked to try higher-fidelity models and test more components. Still, I managed to choose four meaningful components based on user needs and create wireframe-level design strategies.

Moreover, at the start of the project, I didn't have much experience with VR. It took some time to get used to the tools and understand how to design for spatial environments. But over time, I gained valuable hands-on experience and became more confident in designing for XR.

Overall, this process helped me see how important it is to involve users early, listen to their feedback, and stay open to change. Designing with users, not just for them, made a big difference.

9 Conclusion

This thesis explored how Philips' Design Language System can be meaningfully extended into XR environments by integrating insights from users, designers, and developers. Using a mixed-method approach combining card sorting activities with Philips stakeholders and co-design workshops with end users helped identify which existing 2D DLS elements translate well into spatial contexts, and where new interaction paradigms are needed.

Four new XR-native components were developed based on real-world interaction challenges and creative ideation with participants: the Radial Menu, which enables intuitive gesture-based tool access; the Gaze-Aware Tooltip, which presents contextual information based on user attention; the Navigation Assist Panel, offering spatial orientation support; and the Gesture-Onboarding Overlay, which eases the learning curve for gesture-based systems. These designs were supported by layout heuristics and accessibility principles drawn from literature, usability feedback, Philips' DLS, and market research across major XR platforms.

The resulting XR UI Guide serves as a bridge between Philips' robust 2D design system and the evolving needs of immersive environments. It directly reflects priorities emphasized during the research, including accessibility, modularity, context awareness, and consistency. Future research should assess whether the guide supports early testing and successful integration of XR design principles into Philips' ecosystem.

Although further validation through higher-fidelity prototyping and usability testing is still needed, this work lays a strong foundation for a scalable and user-centered spatial design system. Ultimately, this thesis demonstrates that careful user research and systematic design adaptation can help extend trusted design systems into XR in a way that is thoughtful, inclusive, and technically feasible.

10 References

[1] K. Todi and T. R. Jonker, “A Framework for Computational Design and Adaptation of Extended Reality User Interfaces,” *arXiv preprint arXiv:2309.04025*, 2023.

[2] M. Huzaifa *et al.*, “ILLIXR: Enabling End-to-End Extended Reality Research,” in *Proc. IEEE Intl. Symp. on Workload Characterization (IISWC)*, 2021, pp. 52–64.

[3] V. A. Vieira *et al.*, “Understanding affordances in XR interactions through a design space,” in *Proc. IHC 2024 – Brazilian Symposium on Human Factors in Computing Systems*, Oct. 2024, pp. 1–14.

[4] S. A. Ghasempour Yousefdeh and S. S. Oyelere, “Investigating Co-Presence and Collaboration Dynamics in Realtime Virtual Reality User Interactions,” *Frontiers in Virtual Reality*, vol. 5, Article 1478481, 2024.

[5] J.-E. Shin *et al.*, “The Effects of Device and Spatial Layout on Social Presence During a Dynamic Remote Collaboration Task in MR,” in *Proc. IEEE ISMAR*, Oct. 2022, pp. 399–410.

[6] L. Wu and Y. Zhi, “Extended Reality in Language Learning: A Cognitive Affective Model of Immersive Learning Perspective,” *Frontiers in Psychology*, vol. 14, Art. 1109025, 2023.

[7] H. Luo *et al.*, “Integrating Inquiry-Based Pedagogy with Mixed Reality: Theories and Practices,” in *Proc. 4th Intl. Conf. on Computer Science and Technologies in Education (CSTE)*, 2022, pp. 82–86.

[8] S. Oh, H. J. So, and M. Gaydos, “Hybrid Augmented Reality for Participatory Learning: The Hidden Efficacy of the Multi-User Game-Based Simulation,” *IEEE Trans. Learning Technologies*, vol. 11, no. 1, pp. 115–127, 2018.

[9] J. W. Morgan, R. A. Patel, and S. Campbell, “Practical Considerations of Clinical XR (AR/VR) Deployments,” *Frontiers in Virtual Reality*, vol. 6, Art. 1517402, 2025.

[10] N. E. Seymour, A. G. Gallagher, S. A. Roman, M. K. O'Brien, V. K. Bansal, D. K. Andersen, and R. M. Satava, "Virtual reality training improves operating room performance: Results of a randomized, double-blinded study," *Annals of Surgery*, vol. 236, no. 4, pp. 458–464, Oct. 2002, doi: 10.1097/00000658-200210000-00008.

[11] Philips, “LumiGuide – 3D device guidance (NCVC933),” Philips USA, 2025. [Online]. Available: product page for NCVC933 LumiGuide 3D Device Guidance :contentReference[oaicite:1]{index=1}

[12] T. Arthur, G. J. Melendez-Torres, D. Harris, S. Robinson, M. Wilson, and S. Vine, “Extended Reality Interventions for Health and Procedural Anxiety: Panoramic Meta-Analysis Based on Overviews of Reviews,” *Journal of Medical Internet Research*, vol. 27, p. e58086, Jan. 2025.

[13] P. Dhunnoo, L.-C. Wetzlmair, and V. O'Carroll, "Extended Reality Therapies for Anxiety Disorders: A Systematic Review of Patients' and Healthcare Professionals' Perspectives," *Sci*, vol. 6, no. 2, p. 19, Apr. 2024, doi: 10.3390/sci6020019.

[14] R. Holopainen, J. Tiihonen, and M. Lähteenvuo, "Efficacy of immersive extended reality (XR) interventions on different symptom domains of schizophrenia spectrum disorders: A systematic review," *Frontiers in Psychiatry*, vol. 14, p. 1208287, Aug. 2023. doi: 10.3389/fpsyt.2023.1208287.

[15] V. R. Curran and A. Hollett, "The use of extended reality (XR) in patient education: A critical perspective," *Health Education Journal*, vol. 83, no. 4, pp. 338–351, 2023. doi: 10.1177/00178969231198955.

[16] Philips Healthcare, "Pediatric Imaging Experience," *Philips.com.sg*, 2021. [Online]. Available: https://www.philips.com.sg/healthcare/technology/pediatric-imaging-experience

[17] Philips, "Philips launches Pediatric Coaching to enhance MR imaging patient experience for young children," *Philips News Center*, Oct. 11, 2021. [Online]. Available: https://www.philips.com/a-w/about/news/archive/standard/news/press/2021/20211011-philips-launches-pediatric-coaching-to-enhance-mr-imaging-patient-experience-for-young-children.html

[18] F. Loizides *et al.*, "Implications of Immersive Technologies in the Healthcare Sector and its Built Environment," *Frontiers in Medical Technology*, vol. 5, 2023.

[19] Philips Experience Design, "Filament Design System," Philips, Accessed: Jun. 21, 2025. [Online]. Available: https://filament.philips.com

[20] Philips Design, "Showcase: HoloLens," *Philips Design Language System*, [Online]. Available: https://www.dls.philips.com/digital/showcases/showcase-hololens.

[21] Apple Inc., "Human Interface Guidelines: visionOS," Apple Developer, 2024. [Online]. Available: https://developer.apple.com/design/human-interface-guidelines/visionos/overview

[22] Android Developers, "XR UI Design," *Android Developer Guide*, Accessed: Jun. 21, 2025. [Online]. Available: https://developer.android.com/design/ui/xr

[23] Meta, "UI Set | Meta Horizon OS Developers," *Meta Developer Portal*, Accessed: Jun. 21, 2025. [Online]. Available: https://developer.oculus.com/documentation/horizon/os-ui/

[24] Microsoft, "Mixed Reality Design Language (MRTK3 UX Components)," Microsoft Learn, 2024. [Online]. Available: https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk3-uxcomponents/packages/uxcomponents/mixed-reality-design-language

[25] S. Davari and D. A. Bowman, "Toward Context-Aware Adaptation in Extended Reality: A Design Space for XR Interfaces and an Adaptive Placement

Strategy,” arXiv preprint arXiv:2311.03988, Nov. 2024. [Online]. Available: https://arxiv.org/abs/2311.03988

[26] R. Bovo, S. Abreu, K. Ahuja, E. J. Gonzalez, L.-T. Cheng, and M. Gonzalez-Franco, “EmBARDiment: an Embodied AI Agent for Productivity in XR,” arXiv preprint arXiv:2408.12345, Aug. 2024. [Online]. Available: https://arxiv.org/abs/2408.12345

[27] P. Dourish, *Where the Action Is: The Foundations of Embodied Interaction*, MIT Press, 2001.

[28] K. Pfeuffer, J. Alexander, M. T. Marshall, and H. Gellersen, “Gaze+Gesture: Expressive, Precise and Targeted Interaction with Smart Glasses,” in *Proc. of the ACM CHI Conference on Human Factors in Computing Systems (CHI '17)*, Denver, USA, 2017, pp. 3018–3029. doi: 10.1145/3025453.3025738

[29] A. Zaragoza, *XR/ VR UI Kit*, Figma Community, 2024. [Online]. Available: https://www.figma.com/community/file/1253443272911187215

Disclaimer

This thesis involved using AI-based tools to support parts of the writing and research process. Grammarly was used to check grammar and improve clarity. OpenAI’s ChatGPT was used for polishing text, clarifying structure, and generating phrasing. Elicit was used during the literature review phase to help summarize academic sources, to filter and read only the most relevant papers, all of which were reviewed and cited appropriately. All ideas, analyses, design decisions, and conclusions presented in this thesis are the original work of the author. The tools were used in a non-generative, assistive manner and by academic integrity standards.

All platform-specific design patterns referenced in this thesis are derived from publicly available developer documentation and are used for academic analysis.

11 Appendices

Appendix A. Consent Form

Consent Form for Developing a User-Centered Design Language Systems for XR Environments

YOU WILL BE GIVEN A COPY OF THIS INFORMED CONSENT FORM

Please tick the appropriate boxes	**Yes**	**No**
Taking part in the study		
I have read and understood the study information dated / / , or it has been read to me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	☐	☐
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.	☐	☐
I understand that taking part in the study may involve 1-3 of the following:	☐	☐

- Participating in an **interview**, which will be **audio-recorded/video-recorded and transcribed** for analysis.
- Taking part in a **usability testing session**, where I will interact with a product or system while my feedback and actions are observed and recorded via **audio/video/written notes**.
- Participating in a **co-desing session**, where my responses will be documented.
- **Providing feedback** on my experience during the session.

Risks associated with participating in the study		
I understand that taking part in the study involves the following risks:	☐	☐

- Possible **mental discomfort** when answering interview questions.
- Potential **frustration or fatigue** when interacting with the tested product/system.
- Risk of **personal identity being revealed** in group discussions or recordings.

Use of the information in the study		
I understand that information I provide will be used for:	☐	☐

- **Academic research and analysis** related to usability and user experience.
- **Reports, presentations, or publications** based on findings from the study.
- **Possible improvements to the tested system/product.**

I understand that personal information collected about me that can identify me, such as [e.g. my name or where I live], will not be shared beyond the researchers and employees involved in the research team.	☐	☐
Consent to be Audio/video Recorded		
I agree to be audio/video recorded.	☐	☐

UNIVERSITY OF TWENTE.

Future use and reuse of the information by others

I give permission for my **anonymized data** (e.g., transcripts, usability testing observations, questionnaire responses) to be archived and used for future research and learning. ☐ ☐

I give the researchers permission to keep my contact information and to contact me for future research projects. ☐ ☐

Signatures

Name of participant: ______________________

Signature: ______________________________

Date: ___________

I have accurately read out the information sheet to the potential participant and, to the best of my ability, ensured that the participant understands to what they are freely consenting.

Name of the researcher: ______________________

Signature: ______________________________

Date: ___________

Study contact details for further information: y.eryilmaz@student.utwente.nl

Additionally, you can contact the Secretary of the Ethics Committee of the Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente through ethicscommittee-cis@utwente.nl

UNIVERSITY OF TWENTE.

Appendix B. Participant Information Sheet

Study Title: Developing a User-Centered Design Language Systems for XR Environments

Researcher: Yağmur Eryılmaz
Institution: University of Twente
Contact Email: y.eryilmaz@student.utwente.nl

Purpose of the Research

This research explores how **Design Language Systems (DLS) can be adapted for Extended Reality (XR) environments** to improve usability, accessibility, and consistency in immersive interfaces. The study will involve **interviews**, **usability testing and co-design sessions** to gather insights on existing design principles and how they can be effectively translated into 3D environments.

Benefits and Risks of Participating

Benefits:

- You will contribute to the **development of improved design standards for XR interfaces**.
- Your input will help create **more accessible and user-friendly immersive experiences**.
- You will gain firsthand experience with **XR prototyping and usability evaluation techniques**.

Risks and Mitigation Strategies:

Potential Risk	Mitigation Strategy
Physical Discomfort: Extended use of XR devices can cause **motion sickness, eye strain, or fatigue**.	Sessions will be **limited to 30 minutes**, and participants can take **breaks or stop at any time**.
Cognitive Load and Fatigue: Testing multiple interface designs may be mentally demanding.	Activities will be **paced to avoid overload**, and clear instructions will be provided.
Privacy Concerns: Participants may worry about their data being linked to their identity.	All collected data will be **fully anonymized**. No personal identifiers will be stored.
Recording Concerns: Some participants may feel uneasy about being video- or audio-recorded.	**Participation in recording is optional**. Notes will be taken as an alternative.
Unfamiliarity with XR Technology: Some participants may find XR interactions difficult or confusing.	**Onboarding sessions** will be provided to ensure participants feel comfortable with the technology.
Withdrawal Concerns: Participants may worry about their ability to withdraw after providing data.	Participants can **withdraw at any time**, and their data will be **deleted upon request**, even after the session.

What Participation Involves

The participant will be asked to participate in 1-3 sessions of the following;

Interview (20–30 minutes): With your consent, the interview will be audio-recorded for accurate transcription and analysis and will be conducted **[in-person/online]**. This will provide feedback on exploring design language systems adaptation for 3D environments

Usability Testing (20–30 minutes): You will interact with **prototypes of XR interfaces** and provide feedback on usability, design, and accessibility.

Co-Design Sessions (30–45 minutes): You will participate in a **collaborative design exercise**, where you will suggest improvements and discuss potential design solutions. The session will take place **in person/online (depending on availability)**, and you will have the opportunity to **express your preferences and concerns throughout the process**.

Procedures for Withdrawal from the Study

Participation is **entirely voluntary**. If at any point you wish to withdraw, you may do so **without any consequences**. Any data collected from you before withdrawal will be **securely deleted upon request**.

Data Collection and Privacy

- The study will collect verbal responses and optional demographic information from interviews, observational data, usability performance metrics, and qualitative feedback from co-design discussions**.**
- **If you consent to recording, audio/video data may be used for analysis**. Otherwise, only anonymized notes will be taken.
- Your personal information will be **kept confidential and anonymized** before any publication or presentation.
- You have the right to **review, modify, or request deletion of your data** at any time.

Data Usage, Confidentiality, and Safeguarding

- The data will be used internally by Philips for industry-oriented purposes, including research and the adaptation of the brand-specific DLS for XR environments.
- No personally identifiable information will be shared outside the research team.
- Findings may be published in **academic journals, conferences, or reports**, but no identifiable information will be disclosed.

Retention Period

Research data will be anonymized within the first 6 months, after which only anonymous data will be retained for a period of up to 5 years to allow for further analysis, potential follow-up research, and academic publication

Contact Information

For any questions or concerns, please contact:
Yağmur Eryılmaz – y.eryilmaz@student.utwente.nl

For ethics-related inquiries, you may contact the EEMCS Ethics Committee:

ethicscommittee-cis@utwente.nl

Appendix C. Interview Guide #1

Phase	Objective	Example Questions
Warm-up	Clarify role, team structure, and context	- Could you describe your role and how your team works with the DLS? - How closely do you work with designers when new components are implemented?
Body – A: Toolkit Usage	Understand adoption and prototyping of components	- How frequently do teams rely on the DLS toolkit? - Are there adoption challenges? - What tools/workflows do you use to prototype DLS components?
Body – B: XR Adaptation	Identify technical barriers or considerations for XR	- What challenges do you see when adapting the DLS to XR environments? - Can current components be extended to 3D, or do they need rethinking entirely?
Body – C: Accessibility & Feedback	Understand how accessibility is maintained and feedback gathered	- How is accessibility ensured across React/Angular? - What testing methods are used? - Who is involved in testing and feedback?
Body – D: System Evolution	Learn how updates, adoption, and migration are managed	- How do teams adopt new DLS versions? - How is a new component or feature added? - What is the timeline for new component rollout?
Wrap-up	Close the conversation, allow follow-up, invite referrals	- Anything else I should consider regarding DLS for XR? - Anyone else I should speak with? - Open to follow-up conversations later on?

Appendix D. Interview Guide #2

Phase	Objective	Example Questions
Warm-up	Build rapport and understand the participant's role in the DLS project	• Could you briefly describe your role and your involvement with the DLS? • What excites you most about working with the design system at Philips?
Body – A: Understanding the DLS	Learn about the principles, structure, and tools used in the DLS	• How does Philips define the core principles of its design language system? • What tools do you use to design and iterate on DLS components? • Can you walk me through the workflow of creating a new component from scratch?
Body – B: Feedback & Testing	Understand how components are validated and maintained	• What testing methods do you use to validate new components? • How do you gather feedback from designers and developers? • What challenges have you faced in scaling the DLS across teams and frameworks?
Body – C: XR Adaptation	Explore perceptions on extending the 2D DLS to 3D/XR environments	• What do you see as the biggest challenge in adapting the DLS to XR? • How do you see design consistency evolving between 2D and XR interfaces? • Are there any DLS components that are especially difficult or easy to adapt to XR?
Wrap-up	Conclude the session and invite follow-up	• Is there anything else you think I should consider about adapting the DLS to XR? • Would you be open to a follow-up conversation later in the process? • Could you recommend anyone else I should speak with?

Appendix E. Interview Transcript # 1

27 February 2025, 01:09PM
21m 42s

Interviewer
Lets dive in and start with the questions.
My first question is how does Phillips define the core principles of its design language system?
How do you decide what the core principles are?

Participant
Principles in which regard?

Interviewer

Like accessibility, usability.

Participant
So let's put this way. The the main business purpose behind the design system is that the organization can create consistent quality fast and cheaper. So in that regard, that's how we also be measured and quality for an organization like Phillips entails that it was conformed with internal external standards.
The total standards is things like brand external standards like visibility and accessibility standards. And. Yeah, I'm not sure that's how we define it

Interviewer

OK, what tools do you use to design and iterate on DLS components?

Participant
You mean design tools?

Interviewer
Yes

Participant

Figma

Interviewer

Yes, just Figma? Because I already saw it. But like, they only use figma for like every framework and everything on the design side.

Participant
Mostly we use some plugins here and there.
Token studio which we'd love to get rid of. Also to fix my if I they cant provide these kind of functionality because I think they are core to their product.
We're exploring for documentation.
Because we're regulated space we need to archive our documentation which figma doing a great job. Also there we tried to convince figma provide these

kind of functionality.
Yeah, besides that, there's some helpful tools that like a plug in for accessibility that team use.
And we use mirro for communication.
But I wouldn’t consider it as a core tool and we could do that also differently.

Interviewer

So you basically have UX designers for the design right?

Participant

Yes.

Interviewer
Can you walk me through the complete workflow of creating a new component from scratch, from identifying the need for it to the final implementation?

Participant
So we differentiate the design system into three layers lets put this way.
So there's the one that's a foundations.
So that's sizes, colours, font like these kind of things.
That we have the component layer on top of the components we have we call Templates and architect. These are compositions of components.
And to create a component from you cannot start.
Basically, we most of the components are based on demands from the business.
Either direct demands or demands that we see in the business and.
So we try to understand and some of the components will be just designed like a button.
Some of the components we need to understand their functionality, they're very specific and they're too specific then we leave it to the business to see this as a good addition to the design system, because other businesses would benefit from it and we would take it up. And usually there's a phase of exploration, a short phase of exploration.
Then we look at the different variations.
Have a design review on them.
If we feel any of the options is good, then we'll throw that detail it out.
And also add documentation to our Hub website where we keep all the documentation accessible.
And then it moves into Participant A’s space and needs to be implemented across the different vehicles. And once that is done, there's a design review implementation.
So Q&A.
The insurance moment, and if that's passed, then it gets released.
Toolkit and the new component itself obviously needs to make use of the existing foundational elements that are already defined. Sometimes we need to

create new tokens or new foundational elements.
What else? We often for some components, we need to sort of exercise the different variants that might exist. Also investigate if it has an impact to the rest of the design systems if there's something that sort of requires updates with other components that need to be taken into account. If it's a breaking change.
Because just in addition. But if it's a breaking change. Then we will wait to release it undtil we do a major release. But between the major releases, we don't have breaking changes.

Interviewer
OK. Actually, I also have a question about the testing methods that you use to validate the new components like automated testing, usability test testing or developer feedback. Like what kind of methods do you use?

Participant
Well the component itself is very difficult to test. So we do accessibility tests through plugin and human review.
And if it's a component that is based on an immediate request from the business, so they need it for their current development. Then the business will also do testings, depending on the business, this might be AB testing. They might be also testing with
customers. Or at a conference in the backroom with customers.
And what we also do is replace components into our templates and those templates will be reviewed by another design team which is called Design Review team.
And we're working on getting all the templates reviewed by usability engineers in the organization. So the component itself is difficult to test.
If you have screens or if you have something that actually provides the use case, then you can see if it halts to the use case

Interviewer
You actually answered my next question.
Yes, it was like do you connect usable testing on the conference themselves or is it more integrated in the product level?
Yes.

Interviewer

What's been the most effective way to gather feedback from designers and developers using the DLS? Maybe you already answered this as well.

Participant

Well Participant A has the developers. Yeah. So the team works as one team.
The design. Basically does the proposal and gets pitched to developers.
With the developers accepted they will get implemented, if not design needs some updates.

Obviously. Or if you sort of create a editorial design system, you don't really know if it will work in the context of the application. So we also get a lot of feedback from the business teams once they use our product. A lot of these feedback is coming through our team itself. Because if a business wants to migrate to our design system, they don't do it on their own. We are supporting that migration.
So we already see if something's not working,or needs to be updated.
So the team itself basically provides feedbacks.

Interviewer

So its like an iterative process. What challenges have you faced in making the deal as scalable across different teams and use cases and maybe also different frameworks?

Participant

The biggest challenge is to convince the business. Because if you're a product owner.
There's a potential benefit of a design system that you can develop faster that you can offload maintenance of the components to the design systems. Because it's a cheaper but on the other side you have an additional dependency on your product. And depending on the experiences or how the person looks at these kind of things. Umm yeah, that might be. Open for it or not. Plus then every business goes through a cycle, right? So they do a new product introduction.
And it goes into maintenance and they do a midlife update to continue the maintenance and then they develop the next generation.
Yeah. and depending on which business you're in, these can be like 3, 5, 10 year cycles. Obviously if someone just introduced a new product, they will not update, they will not migrate to a design system because they have a maintenance phase.
So in that sense you need to be cautious of the context and manage expectations.
to the people that can report to that certain businesses will not be able for migration for some period of time, so you will not be able to get them over. And we don't have a mandate, so we do not need to work by convincing the business that using all design systems and its their own interest.

Interviewer

So the next section will be about extending the DLS to XR. I'm just gonna get some insights and opinion. So from your experience with digital design, what do you see as the biggest challenge in adapting 2D DLS to XR?

Participant

I don’t see any challenges.

Interviewer

You don't? Why not?
Participant

A We have all outdated designs system for XR. Its not with the current design system but with the previous design system.
So if you're in eindhoven, I can connect you to someone who can maybe give you access to it.

Interviewer
Yes, that would be amazing.

Participant
So you can see it. Yeah, I mean. So every new interaction space or paradigm can't just breakdown existing paradigms. That's why we have the desktop metaphor and the recyle bin for throwing away files. Also what I have seen so far XR interactions that they lean strongly the on the two-dimensional expertise that everybody has.
A button doesn't look much different than a button in 2D. It's just projected on a space or it in there. So in that sense, I don't see as a big challenge. What's nice about it is that it's less confined. So obviously everything is confined into a rectangular square currently, if maybe your computer or mobile phone or watch.
Where if you look if you use full 3D space you don't need to constrain everything to rectangular square so you can be much more open with the design.
I can imagine in the clinical space. So I think in the consumer space it's actually easier. As the consumer space is very works a lot with is is already much more editorial. So much more airy in a sense. Works much more with progressive disclosure in a medical sort of professional space. People want to see all the information that's possible and have all functionality as quick as possible accessible so that also fast results in very dense Uis. They are translating dense Uis into a 3D space. I can see some challenges there more than in the consumer applications.

Interviewer
How do you see design consistency evolving in XR compared to traditional interfaces?

Participant
What do you mean by it is an inconsistency, so like transforming 2D to 3D.

Interviewer
Yeah, that's what I meant.

Participant
Well, I think in the in the end it will be variations. It will not be 1 o 1.
Because a colors will be different, spacing will be different.

Like it's perceived colors will be different.
Density will be different.

Interviewer
I mean, they're also like additional components that needs to be considered like to design for like navigation.

Participant
Yeah, exactly. So there will be differences, but I also would not.
I will generally say it only needs to feel like coming from the same origin and really being the same thing, because I don't think that will make sense,

Interviewer

OK. Yeah, actually that was it. These are all my questions. Yes, because you already answered like some. Yeah, I didn't ask some of them. I really appreciate your time.

Appendix F. Interview Transcript #2

February 28, 2025, 12:00PM
Interviewer 0:04
Yes, just a second.
And Yep, I started the transcript.
Yes, OK.
I'm gonna start with my first question.

Participant 0:16
Yep.

Interviewer 0:19
So how frequently do teams at Philips rely on the DLS toolkit?

Participant 0:25
Well, for some, for some teams it's everyday and for some it's only when they need to do updates of course.
So we have actually.
Well, we have multiple toolkits.
There's one for design and one for for code.
For most of our products, designers are full time busy doing designs. If it's not for one product is for the other one.

Interviewer 0:50
Mm hmm.

Participant 0:52
But let's say the design community will use this.
Yeah, almost daily, but certainly weekly.

Interviewer 0:59
Mm hmm mm.

Participant 0:59
And developers will use this obviously when changes need to be made, right?
I mean if not changes need to be made, then they they don't need to use it.
But yeah, it's it's. It's a part of the product, right?
So every time James product you, you you're confronted with?

Interviewer 1:16
So do you think both designers and developers fully adapted or are there any challenges in its usage?

Participant 1:23
Out as many challenges so.
We have to realize that, you know, having a design system is not necessarily a new thing in Philips.
I mean, we've had design systems since.
Before the year 2000, I think.
So it's not strange, it's not new.
What is new is that we have now both a design toolkit and a code toolkit.
So in the past we didn't have this. And so in the past we mainly had.
A design toolkit.
And then the biggest problem we ran into was that there was no corresponding call toolkit.
So what happened is that most development teams had to implement it themselves. You know, over and over again so.
I think we found somewhere around.
20 different kinds of implementations of the same design system across the company.
So two years ago we formed this new department that I'm now leading and we are now in charge centrally of developing the toolkits.
So we do this in one place in the company, so that not everybody has to do this by themselves, which is of course a lot more efficient because we do it once and and closely together with the designers.
So we take care of also the the quality issues and the bug fixing and all that.
However, it's new, it's a new thing, so some of our customers are extremely happy. They've switched to our toolkits.

They started using it, but yeah, Philips is a large company with a lot of legacy software.
So a lot of teams are not able or not willing or it's not the right moment to switch. And so one of my key KPIs is adoption rate.
So yeah, we're slowly going up, but yeah, it's not like the whole company is using it yet.
So that that that's still a challenge and and one of the well, there's several reasons for not yet adopting it.
I mean, if you have a system you want to change your entire UI with a new toolkit.
This is substantial amount of work.
This is not something you do in two weeks, right?
So this will take months.
And.
Yeah. So it's a matter of the right timing. And so if if there is a, let's say, a period starting when they wanna work on the next version of the system, then maybe it's something you can schedule to happen in the time frame.
Because otherwise, nobody's gonna just do this, right?
They have other bug fixes and things to take care of or new features to be built, or so we have to be a bit patient.

Interviewer 4:03
Mm hmm.

Participant 4:09
We'll take several years for people to adopt this and but so far it's going quite well.

Interviewer 4:12
Mm hmm.
Great. Yes. Nice. OK.
So what tools and workflows does your team use to prototype new DLS components?

Participant 4:24
So usually.
It starts with designers working in Figma.

Interviewer 4:30
Mm hmm.

Participant 4:32

Most, I mean we do components right.
Like a button or tool switch.
So most of those components they are made in figma.
They document each state of the component.
So showing the behavior of the component is a bit difficult in in figma because they're prototyping capabilities of figma is a bit limited, but after let's say if the designers made an initial design, then usually we go quite quickly to a code prototype.
For several reasons.
One it's, you know, designer only thinks about the things that designer thinks about and the coder thinks about other stuff. So.
You know, we usually do some rounds of iterations together.
We know.
We sit together.
We see.
See what works and check out the implementation.
See how it feels.
Maybe add an animation here, or maybe change behavior a little bit.
And then we finalize the design.
The design gets documented on our hub.
I've sent you the link.
Maybe you haven't been able to access it, but.

Interviewer 5:32
Not yer, Mm hmm.
Mm hmm mm hmm.

Participant 5:37
And then we implemented in code which is can be quite a tricky thing because we don't need to do it in one technology. We need to do it for the web we need to.
For iOS we need to do it. For Android we need to do it for all the other toolkits as well so.
That's roughly how it goes.
We use.
So figma as a key tool GitHub is a key tool that we we manage the work and we use that for work planning.

Interviewer 6:01
Mm hmm.

Participant 6:04
We also have some of our own software that we have written.

So in Figma we're using a plugin called Figma Studio.
That allows you to better manage all your variables and tokens.
And that stores all of the tokens in a Jason format and that Jason format we are, we have written our own software to convert that into something that iOS or Android or any other toolkit can actually use.

Interviewer 6:27
Mm hmm.

Participant 6:38
Because Jason is not useful for me on on iOS.
So we need to convert that to something else.
So that's a tool we use Pigment Studio we use, we use storybook a lot, which is a tool where you can list all your components in the various States and you can also test them.
We also use a lot of coding tools.
So we have code for static code analysis.
We have tools for.
Visual regressions it's called chromatic.
So basically what that does, it takes a pixel screenshot of each component and then when you make a change does another screenshot and then you can see if the change was there.

Interviewer 7:30
Mm hmm.

Participant 7:35
Yeah. And then we use all of the built pipelines from GitHub to automatically produce all of the the software tools and build packages so that people can consume that.

Interviewer 7:45
So as far as I understand, you don't use any design systems, right?
Like for react for example like carbon or like material UI like an external library.
So the design is solely based on the figma templates, right?

Participant 8:02
No. So for react we use React area which is a product from Adobe and that's called a a headless toolkit.

Interviewer 8:06
Mm hmm. Mm hmm mm hmm.

Participant 8:10
So it mainly provides hooks for creating components, but the look and feel of the components you can do entirely yourself.

Interviewer 8:11
Mm hmm.

Participant 8:22
On iOS, we so iOS we're doing in Swift UI, but Swift UI is very limited in the customization possibilities.

Interviewer 8:30
Mm hmm.

Participant 8:32
So we ended up writing most components ourselves with a little bit of reuse of existing components.
On Android, we're using material because that's the only thing you can do on Android.
And the angular toolkit we're using CDK, which is the official toolkit for Angular.
So yeah, it's it's, it's a bit depends on the technology.

Interviewer 8:59
OK.
I get it.
Yeah. Thank you.
So from your experience, developing a dls toolkit for web frameworks, what challenges do you think would arise when extending a dls to support XR environments?

Participant 9:13
To export XR environments, yeah.
Umm.
Well, I mean, most of our design system has been based on, let's say a desktop based or touch based interfaces, right?
And you are now going to look at XR, which is obviously different.

Interviewer 9:39
Mm hmm.

Participant 9:40
So I would imagine.
Some of the things apply right?
So I mean, you'll still need a button, for example, you still need to display text.

Interviewer 9:47
Mm hmm.

Participant 9:50
But yeah, I I would imagine in augmented reality you're you're trying to keep, let's say your your vision free of many things and only annotate things.
Yeah, maybe call it.
It's more of an annotation kind of interface rather than a full interface where you use the entire screen.

Interviewer 10:12
Yeah.

Participant 10:15
So I would say you need more things.
Maybe you should look at there.
There's a guy in the team called Bart he's busy with.
Annotations for radiology software.

Interviewer 10:28
Mm hmm.

Participant 10:29
Maybe you can get some inspiration from that.
So yeah, I think you would need new things. You can use some of the things that we've done, but.

Interviewer 10:37
Mm hmm.

Participant 10:38
Limited, I would say.

Interviewer 10:40
OK.

Participant 10:41
I would imagine you have many more transparent interfaces than we.

Interviewer 10:44
Yeah. Yeah, exactly.
And that's so many like interaction elements. So like a lot of new components, need to be considered to design something for XR environments I would say.
But yeah, we'll see.
I will.
I will try to figure it out OK.
So how does Philips ensure accessibility across different frameworks like React, Angular?

Participant 11:08
Yeah. So in general, we're trying to stick to the WCAG guidelines.

Interviewer 11:15
Mm hmm.

Participant 11:16
It has various sides to it, right?
So one is purely let's say design based like things like color and contrast.
Those kind of things.

Interviewer 11:22
Mm hmm.

Participant 11:25
So those things we already tried to tackle on a figma level and so the way we take a look at bonds, we already invest my check contrast levels and that kind of things. So that that part is mostly handled in, in, in figma from a technical perspective other.

Interviewer 11:29
Mm hmm.

Participant 11:44

Things are also very interesting, like keyboard navigation or scalable interfaces, or you know.
Being able to use a voice reader voiceover.
But the implementation of that is different per technology.
So the way that works on, let's say iOS is totally different than in the web.
So what we do is.

Interviewer 12:08
Mm hmm.

Participant 12:10
We we for every component we make we we test this.
It's it's a little bit manual testing because there's no real tools to automate this, but so we make sure that you can label the parts of the interface we can, and so in the end our component ends up in in an application, right?
And a lot of the labels that need to be set for, let's say, a voice reader.

Interviewer 12:35
Mm hmm.
Mm hmm.

Participant 12:38
Need to be set by the application, not by us.
So we we only need to provide the possibility to set it rather than than than fill it in.

Interviewer 12:41
OK.

Participant 12:46
So we check that and we have also a guy in the team who specializes in accessibility who, you know, keeps an eye on all of these things.

Interviewer 12:56
Mm hmm mm hmm but.

Participant 12:57
But basically it's per technology different. We have to do this for technology, yeah.

Interviewer 13:03

Yeah. My specialty is also on accessibility. So I would like to ask you some like other questions also about this.

Participant 13:09
OK. Yeah.

Interviewer 13:10
So, like, do you aim to reach AA or triple A standard like how do you like, do you go over a checklist like how do you ensure that?

Participant 13:20
Yeah.
In general, the minimum is is AA and and triple-A if we can.

Interviewer 13:28
Mm hmm.

Participant 13:30
We we have.
We have different use cases, right?
So for example, if we have, if you look at our phillips.com site, this is a public site and could be used by anybody, right?

Interviewer 13:42
Mm hmm.

Participant 13:42
However, if we make something for radiologists, it's a bit of a different story, right?

Interviewer 13:46
Mm hmm.

Participant 13:48
Because you would hope that the radiologist has good eyesight, right?
Because otherwise he shouldn't be a radiologist.

Interviewer 13:56
Mm hmm.

Participant 13:57
Well, where radiologist is not so much about things like contrast.
I mean, it's still important, but I mean it's a person who can see well.
Yeah, it's more important that we have keyboard navigation in place and really optimize the workflow.
Also for example.
Yeah, I don't know if you've ever seen an ultrasound device that we produce.
The ultrasound looks very weird from a if you. If you've never seen one. It looks weird because as all of these 3D ups and downs and buttons have shapes and stuff, this is really because person does this as the the the scanning device in one hand and looks at the patient while he has one other hand at the machine.
Right. So he's basically using it without looking at it.

Interviewer 14:46
Mm hmm yeah.

Participant 14:51
So yeah, everything has a has a context, right?
So that's my point, so.
In that's why we say in general, we want to make sure we do AA for certain application. Actually it doesn't make much sense to go further than AA because the user you're working with is such a specific user that you know does not have any disabilities.

Interviewer 15:01
Mm hmm.
Mm hmm.
Mm hmm.

Participant 15:13
So yeah, and then, you know, we only do the design system, right?

Interviewer 15:15
OK.

Participant 15:17
We don't do the application, so it's a little bit. I think you had another question about that later on about usability, but accessibility only starts being meaningful in the context of a certain task, right?

Interviewer 15:19
Mm hmm.
Mm hmm.
Mm hmm mm hmm.

Participant 15:31
So often on the base of a component, you can't really say that much other than I've enabled it, it can be said whether it's meaningfully set, we don't know.

Interviewer 15:43
OK.
So what testing methods do you use to validate new components?

Participant 15:51
Well, mostly manual.

Interviewer 15:53
OK, so now OK.

Participant 15:54
Yeah.

Interviewer 15:56
So no usability testing or like maybe developer feedback.

Participant 16:02
Yeah. Yeah, no.
So I mean, so the developer and the designer together they work on it. When the developers made something to go sit together and they test it and see if everything looks right, like like a snapshot.
See if it's pixel perfect.

Interviewer 16:17
So maybe use cases like following use cases and see if there are problems that kind of testing.

Participant 16:22
Yeah, but I mean, what kind of things we wanna test on a button and how can I press the button?

Interviewer 16:26
Yeah, that's true. That's true.

Participant 16:27
That's about all we can do, right? So.
We are very often not talking about even screens.
I mean, we're now moving into the level of screens.

Interviewer 16:36
Mm hmm.

Participant 16:37
But on a component this is usually quite basic.
I mean component does one or two things and that's it. Yeah, so.

Interviewer 16:43
OK.
Do you involve end users or just developers and designers ?

Participant 16:50
We may say all involved developers and designers, what we do in Phillips in general is that each business does usability testing on their their applications.

Interviewer 16:56
Mm hmm.
Mm hmm.

Participant 17:03
So we whenever there's something wrong with the component, we will get feedback based on those kind of.
Studies. But then again, I have to realize Design is is not new for us and so every design thing that we've made since 2000 is basically evolution of the previous one, so.

Interviewer 17:20
Mm hmm mm.

Participant 17:24
We're we're we're evolving, right?
We're not, we we didn't sit like, hey, let's start from scratch and do something

else.
You know, so we know all the basic stuff. We know all the basic behavior that a component is up. Actually, if you look at our design system, to be honest, it's about as middle ground as standard as you can imagine, right?
It's very bare bones.
It's very in your face.
What we are doing now is, like I said, we are now moving to the level of pages or templates as we call them. So for example, a login page or a login flow. We are now starting.
To design these entire flows with all the error handling and all these other things with it.
And we are now working with usability.
Center in Phillips to get those tested and validated, because then it becomes meaningful, right?

Interviewer 18:19
Mm hmm.

Participant 18:21
Because it's an actual flow with an actual task that you can actually evaluate.

Interviewer 18:27
Mm hmm.

Participant 18:27
So we are moving in the direction, but for individual components like a double switch, yeah, there's not much you can do in terms of usability testing.

Interviewer 18:37
OK.
I think you already answered my next question because it was like about feedback from designers and developers.

Participant 18:43
Yeah.

Interviewer 18:44
So I'm gonna skip that one.
OK. The next question is when a new version of DLS is released, how do you handle adoption and migration within the organization?

Participant 18:58

Yeah. So introducing a whole new DLS is is of course complicated.
So I think since 2000 we've done 4.
And that's really complicated because then you really need to do the marketing.
You need to inform everybody. You need to teach them.
So we literally have a communication program where we do all kinds of activities.

Interviewer 19:19
Mm hmm.

Participant 19:23
We do.
We do post on the Internet.
We do presentations.
We call everybody.
We organize and design system day. We do workshops, everything you can imagine.
We try to do as people know, but it's basically a marketing exercise.
Now, technically speaking.
There's also a challenge because.
Like I said, you know, if you have a code base.
In a different technology, you will not easily switch. However, some of our clients are on older version versions of our toolkit. If that is the case, we're trying to make the migration easy by providing.
Migration guides and will help them how to do it.
Sometimes we also do it for our customers, so we do a couple of sessions. We show them how to do it.

Interviewer 20:13
Mm hmm.

Participant 20:19
Or at least get them started in doing it so that the rest they can do themselves.
But yeah, it's a it's a big deal.
It's a big deal.

Interviewer 20:28
Mm hmm, yeah.
So how does Philips decide when to create a new product or feature within the DLS?

Participant 20:38
Yeah, this is very.
Basic. So I mean we don't create really products.
I mean, we have toolkits which are a product.

Interviewer 20:46
Mm hmm.

Participant 20:48
So right now we're seeing, I mean, you know how it is right in, in, in IT land. There's so many technologies and even though we do so if you want Jetpack compose and there's always somebody who wants to do flutter or react native or you know this you.

Interviewer 21:02
Mm hmm.

Participant 21:04
Can't keep up with all of them, so we've decided to do only some technology stacks.

Interviewer 21:05
Mm H.
Mm hmm.

Participant 21:12
If people want to do something else, we simply say sorry, we're not supporting you.

Interviewer 21:16
OK.
So do companies ask you to create something for themselves specifically or not?

Participant 21:24
No. So the way it works is that we have quite a lot of components now and so it's about I know 60 plus components. So most of the basic stuff is there.

Interviewer 21:32
Mm hmm.

Participant 21:36
But regularly it does happen that we get requests.
So either for modifications to existing components.
So for could be something very simple like. You know, I would like to be able to set a different font size for a link component because we have a standard size that we use.

Interviewer 21:50
Mm hmm.

Participant 21:53
But sometimes there's a reason to have a different one.
So then we add that possibility.

Interviewer 21:58
Mm hmm.

Participant 21:58
Sometimes they ask for new components and say like, well, we have component you don't have and we think it's useful also for others. I have to have because we don't make any time you make an application, you always gonna have custom components.
Yeah. So any application would maybe consist 70% out of our components and 30% out of components that they make themselves because they're application specific. And so some of them could be generic and useful for other businesses.
So then we we take on the request.
We we plan it.
We design another component, we add another one, but to be honest, doesn't happen that much.
I think in the whole of last year we maybe added four components or something like that.
So it's not.

Interviewer 22:40
Mm hmm mm hmm.

Participant 22:41
It's not a huge amount.
Most of the feedback we're getting are, you know, can you add a parameter to this component that does this or can you add another variant of this

component? Or, you know, there's accessibility issue with this component or you know it doesn't scale properly or or it doesn't scale like I wanted to have it scale because some of these things, especially in the area of accessibility for example on mobile phones you have this something called dynamic typing. So in your phone settings you can increase the phone size, but then there's a question what happens, you know.
So let's say you have a label.
Increase the font size.
Does everything around it also get bigger or does does it get truncated?
Does it wrap over 2 lines?
Some of these things are a little bit subjective, so sometimes we end up in discussions like, you know, one person thinks it should be this way. The other person thinks it's that way and then, well, we'll go through the motions, try to resolve it.
It's more that kind of feedback.

Interviewer 23:38
Mm hmm.

Participant 23:39
It's not so much the new components.

Interviewer 23:43
OK.
How long does it typically take to develop and implement a new DLS component from ideation to release?

Participant 23:55
Yeah, it could be anything from one day to a month.

Interviewer 24:00
Depends on the component.

Participant 24:01
Then, based on the complexity, so for example.

Interviewer 24:04
Thank you.

Participant 24:06
Let'd say a link component could be in one day because it's so simple.

Interviewer 24:10
Mm hmm.

Participant 24:10
Whereas for example a data grid or a date picker could take up to a month depends on the complexity.

Interviewer 24:21
OK.

Participant 24:23
Or maybe on on average, two weeks or so, three weeks.

Interviewer 24:23
So.
OK.
So is there anyone else at Philips that you think I should speak with to get more perspective?
You already mentioned someone but like if there anyone else that I can reach out to during the design process also?

Participant 24:36
Yeah.

Interviewer 24:42
That will be really useful for me.

Participant 24:51
I don't know.
I mean, I I think you're already with Participant B in a good place.
Because I I think your challenge is not so much about design systems, it's more about what are the components you need in in augmented reality. Umm. Yeah. I I I think Participant B knows better than me. Who else in the company has experience with augmented reality.
Because I I don't have much experience with it.
For any design system related question, you can ask our team, but yeah.

Interviewer 25:26
Mm hmm. OK.

Participant 25:28
Don't know.
We'll let you know if something comes to mind.

Interviewer 25:31
OK.
Thank you so much.
This has been really valuable.
That was it.

Participant A 25:35
OK, cool.

Interviewer 25:36
These are all my questions.
Yes, thank you so much for your time again.
Do you wanna add something?

Participant 25:43
Now, do you have already some initial ideas of where where you think you'll be headed?

Interviewer 25:48
Yeah. I'm gonna, like, focus on the navigation inter interaction elements.
So basically navigation within XR like some UI components that user can actually engage with.
So I I wanted to narrow it down a little bit because it's already really complicated and I'm gonna study the DLS first, the existing one and see if I can transform some of the 2D components to 3D.
With plugins for example, I already found that like in figma there's this unity plug in.
I'm gonna try it with that, but I don't know if if it's like good enough, you know. So I'm gonna look for like other maybe tools to use.
That's my start point.

Participant 26:32
Maybe some some random comments here, but personally I for Phillips in the context of Phillips I already saw always saw augmented reality as something that would be particularly useful in the area of, let's say operations on a patient, right?
Because right now.

I think Participant B can show you, but many of our systems are based on the idea that the surgeon has his hands free.
To to work on the patient. And then usually we have like big panels in front of the the the surgeon with all kinds of information about what he's doing.

Interviewer 27:08
Mm hmm.
Mm hmm.

Participant 27:13
So we we call this guidance, right and of course augmented reality could be one interesting way of doing guidance because you know you instead of looking at a screen which is fixed in same position, you could show information on the patient where the patient actually is, right, so.
A lot of it will be about actually not interacting at all, right?
So just using added as a as a way to see things that you could otherwise not see very easily because the the the surgeon will have his hands quite busy doing other stuff.

Interviewer 27:47
Mm hmm.

Participant 27:51
So he has very little time for navigation and it's more about guiding him in in what he's doing.

Interviewer 27:57
Mm hmm.

Participant 28:01
But it's just a thought.
Think about it.

Interviewer 28:03
No, no, it's it's nice.
Thank you. Yeah.

Participant 28:06
Yeah, no.

Interviewer 28:07

I will think about it here.
I'm still trying to mold my idea.
about the thesis, but it's getting together. We'll see.

Participant 28:15
OK.
Yeah, OK, well, good luck.

Interviewer 28:17
Thank you.
Thank you so much.
Yeah. Have a nice day.

Participant 28:20
No worries.

Have a good weekend. Bye.

Appendix G. VR Simulation User Feedback Summary

#	Overall Experience (1-5)	Ease of Use: Interaction & Understanding	Realism / Relevance: Suggestions for Engagement
1	5	Setup took time, no interaction, wire unreachable, frustrating	Add onboarding, interaction; currently feels like a passive video
2	5	Hard to reach numbers, unclear interaction, hand usage confusing	Clarify user actions, improve interactiveness
3	5	Natural, responsive, strange tactile sensation	Interactivity good, but physical feedback lacking
4	3.4	Grabbing nobes was difficult; 2nd simulation more familiar	Make nodes more functional, improve realism

5	4	Technically easy, wanted something unique	Add a patient story, include a nurse role
6	3	Hard to navigate, lack of tactile feedback	Needs realism, clearer controls
7	5	Easy to use, intuitive	Add sound and tactile feedback, improve text clarity
8	4	Clear image, decent interaction	Focus on communication cues, polish interface
9	5	Intuitive but controller was confusing	Add controller feedback, fix finger positioning
10	4	Confused about grabbing machine, small space	Tactile feedback, gravity, larger area requested
11	5	Responsive but delay confusing	Add physical interaction, remove distractions
12	4	Immersive, interaction OK, needed precise pickup	Include patient and doctor visuals, AR glasses
13	4	Unfamiliar with system, no feedback	Add feedback for interaction confidence
14	4	Handles unclear, needed guidance	Better onboarding for new users
15	5	Responsive, easy to understand	Prevent interaction cut-off from external events

16	5	Very enjoyable, intuitive	Add more voice-over, distinct from passive video
17	5	Confused by how to interact	Clarify gestures, grabbing logic
18	5	Realistic, but dizziness on entry	Improve alignment and smooth transition
19	3	Pinching/grabbing failed	Add clear hand gesture instructions
20	4	Letting go and re-selecting was confusing	Improve gesture tracking logic
21	4	Gesture confusion, no haptics	Add haptic feedback, improve system response
22	4	Took time to learn, joystick not intuitive	Use real-world size, add speed control
23	4	Pressing unclear, not precise	Show where touches register
24	5	Realistic, manipulations not intuitive	Improve onboarding, resolution
25	5	Hand syncing needed, struggled with controls	Include a mini demo, onboarding, system check

Appendix H. Focus Group/Co-design Workshop Presentation Link

The presentation used in the session can be accessed at:

https://www.canva.com/design/DAGlcfcZFUY/IgzJuFE9y9Ts1eY1EvuUgQ/view?utm_content=DAGlcfcZFUY&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utlId=h44582c628b