Xocchiali AR glasses

Design, Implementation and usability testing of simple AR glasses

Adam Korytowski, Aleksy Królczyk

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2023

MASTER THESIS

Xocchiali AR glasses

Design, implementation and usability testing of simple AR glasses

Adam Korytowski, Aleksy Królczyk

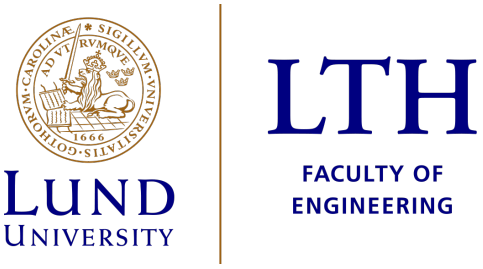
Xocchiali AR glasses

Design, implementation and usability testing of simple AR glasses

Published by

Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O. Box 118, SE-221 00 Lund, Sweden

Supervisor: Günter Alce
Co-supervisor: Paul Johansson
Examiner: Mattias Wallergård


LUND
UNIVERSITY
LTH
FACULTY OF
ENGINEERING

Abstract

This thesis focuses on designing and prototyping a companion application for Xocchiali's AR glasses which are to be stylish, unobtrusive and cheap smart glasses designed to be used on a daily basis. The main goal was to create a user-friendly application that helps configure the glasses for everyday use. Additional goal was to explore possibility of implementing designed features on a microcontroller that is to be used in the glasses. Key research focus on suitable use cases for simple smart glasses and effective mobile application design. The thesis presents low and high fidelity prototypes of the companion iOS application that guides the user and allows him to configure the glasses. In order to evaluate the usability of the high fidelity prototype, a study was conducted. Results suggest that developed design appeals to the users and is easily understandable, allowing for seamless and satisfying configuration of the glasses. The authors conclude that both the prototype as well as microcontroller application proof of concept could used as a good foundation for the actual, final product.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

As it stands for today, many of the augmented reality (AR) glasses available on the market, despite having many capabilities, are not really practical for a casual user in everyday life [1, 2]. The main reasons for that are as follows:

- Their shape is often bulky and uncomfortable when wearing for longer periods of time.
- The price tag is unreasonably high from a casual person's point of view.
- Battery life is rather short.
- Interaction with them is usually unnatural and unintuitive; also might look weird for passers-by.
- Their design is very distinctive and outstanding compared to traditional glasses, making them unattractive for the general audience.

Due to these reasons, AR glasses' applications are limited to commercial use. Xocchiali aims to change this by introducing affordable AR glasses that are both minimalistic and stylish, allowing them to be worn on a daily basis without drawing unnecessary attention. In return, however, the capabilities of the device will be greatly reduced compared to current devices like Microsoft HoloLens. Importantly, unlike other AR glasses available, Xocchiali's glasses should be designed to be worn all the time so a lot of emphasis will be put on optimizing the power consumption to ensure long-lasting use.

Xocchiali plans to provide the following means for interaction in their glasses:

- A bone conduction system for audio

- An LED light
- A Microphone
- A button

1.1 Purpose and goals

Since the interaction capabilities of the glasses are scarce, additional mobile, companion applications are expected. It should help the user with using the glasses as well as allow them to configure how the glasses behave.

The main goal of this master thesis was to design, prototype, implement and test the mobile application that will be used along with the aforementioned glasses. What such an application would do is not really specified apart from the two main goals - they should lead seamlessly the end users through connecting the glasses for the very first time (what we will call **onboarding process** from now on) and allow them to configure all the necessary things to make their glasses usable and practical.

Another goal of this master thesis is to work with a Qualcomm microprocessor that is aimed to be the target chip to be put in the final glasses. By work, we mean implementing communication between the microprocessor and the mobile phone via the Bluetooth protocol.

1.2 Research questions

With this purpose and goal in mind, we posed the following research questions that we aimed to answer:

1. What use cases are suitable for simple smart glasses?
2. What would be an appropriate and usable design of the mobile phone application for interaction with AR glasses?

1.3 United Nations Sustainable Development Goals

The UN's Sustainable Development Goals (**SDGs**) are a collection of 17 global goals aimed at promoting economic, social and environmental sustainability. Xocchiali's AR glasses, with their focus on minimalistic design and practical applications, can contribute to these goals in several ways:

1. Affordable and Clean Energy (Goal 7) - Xocchiali's AR glasses optimize power consumption, promoting efficient use of energy resources and reducing carbon emissions.
2. Industry, Innovation, and Infrastructure (Goal 9) - by offering affordable, accessible AR glasses, Xocchiali fosters inclusive technological advancement and sustainable industrialization.
3. Responsible Consumption and Production (Goal 12) - focusing on essential functionalities, Xocchiali's AR glasses promote responsible digital consumption, device longevity, and reduced electronic waste.
4. Climate Action (Goal 13) - the glasses' focus on low power consumption contributes to reduced greenhouse gas emissions, indirectly supporting climate change mitigation efforts.

1.4 Limitations

Unfortunately, during the time of writing this thesis the glasses themselves were not released yet. As a consequence, most of the focus was put into the parts of the application that do not require a direct interaction with the glasses.

Chapter 2

Theoretical background

The following chapter describes related research as well as important concepts and terms regarding both design theory and computer science theory as are used later throughout the thesis.

2.1 Related research

2.1.1 Adoption of AR glasses

As it stands today, we can say for sure that AR glasses, among other wearable devices, are getting more and more popular each year. It is also estimated that headwear and eyewear are going to be the second-largest and second-fastest-growing (after handwear) segment of wearables by 2030 year [3]. However, it seems that most of the research done in regards to how well adopted AR glasses are (or what hinders this adoption) concern industrial, not everyday use.

For example, Smith et al. [2] focus on logistics and conclude that more work needs to be done in terms of how comfortable today's solutions are as users point out that they feel discomfort even after short sessions and that the weight distribution of the device is an issue. It is also stated that another obstacle may be generation gap and learning curve associated with new high-tech devices.

In another study called "What will it take to adopt smart glasses: A consumer choice based review?" done by Basoglu et al. [1] the authors based on gathered data state the following, among others, prepositions:

1. Price influences the adoption of smart glasses - lower cost will naturally lead to being more popular. Unfortunately, AR glasses, being still relatively young and complex technology, are still often expensive.

2. Users prefer standalone devices - users seem to prefer devices that do not require syncing with, for instance, mobile phones. They authors point out, though, that this preference may change as users become familiar with the technology.

2.1.2 Augmented reality & Design theory

Augmented Reality

Back in 1994, as more and more concepts and devices emerged in, what we would call today, AR/VR field, Paul Milgram and Fumio Kishino saw the necessity to clearly define different concepts for the academic world and more. They, among others, defined what is today often referred to as *Milgram's Reality-Virtuality Continuum* (see Fig. 2.2) [4] that is up to this day a premise upon which new concepts are based.

Figure 2.1: Milgram's Reality-Virtuality Continuum

Source: "A Taxonomy of Mixed Reality Visual Displays" [4]

Augmented reality (AR) is defined as a real-time view of a physical, real world environment that has been *augmented* by adding computer-generated images and information on top of it [5]. On the continuum shown above, it is clearly closer to the real environment rather than the virtual environment.

Another, commonly used definition by Ronald T. Azuma describes augmented reality as a any system that has the three following properties [6]:

1. It combines real and virtual
2. It is interactive in real time
3. It is registered in three dimensions

User-centered Design

User Centered Design (often abbreviated as UCD) is a design approach that focuses on needs, wants and limitations of the end-users. It relies on the end-users themselves during the design phase, as they are tested and their feedback and input is taken into account in the next phases of development. It is worth noting that due to its nature UCD is an example of an *iterative process*[7].

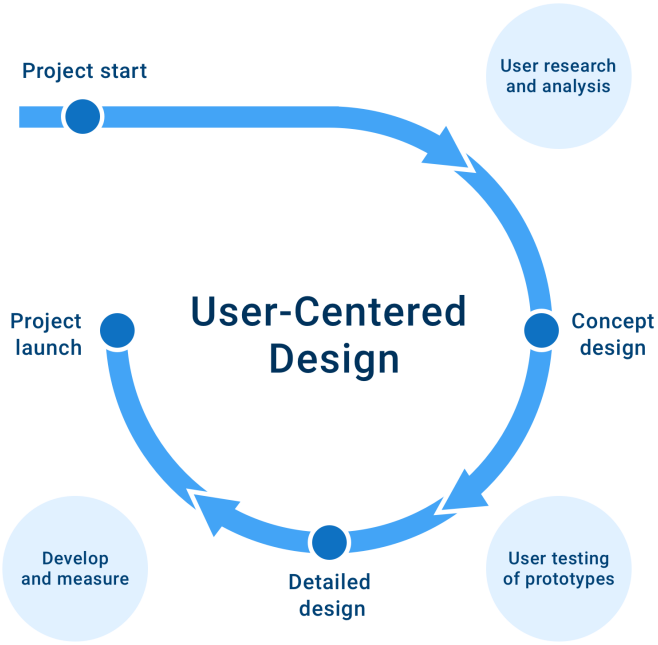


Figure 2.2: Depiction of User-Centered Design process

Source: https://marketsplash.com/user-centered-design/

Usability

Usability refers to the ease and effectiveness with which users can interact with a product, system, or interface, be it digital or physical. It encompasses factors such as learnability, efficiency, memorability, error tolerance, and user satisfaction. A strong focus on usability ensures that a design is accessible, intuitive, and streamlined, empowering users to achieve their goals with minimal effort or frustration.[8]

Usability testing

Usability testing is a crucial aspect of evaluating the effectiveness of a user interface or product design by observing real users as they interact with the system or product. This evaluative method focuses on the ease of use, efficiency, and overall user satisfaction. It aims to identify potential usability issues, user preferences, and areas for improvement, ultimately leading to a more intuitive and user-centered design. Usability testing often employs various techniques, such as task-based scenarios, think-aloud protocols, or heuristic evaluations, to gather actionable insights into user behavior and expectations. By conducting usability tests at various stages of the design process, designers and developers can iteratively refine their products, ensuring that the final outcome aligns with user needs and delivers a seamless, enjoyable experience.[8]

User Experience

User Experience (UX) refers to the overall experience a person has when interacting with a product, system, or service. It encompasses all aspects of the user's interaction, including their emotions, perceptions, behaviors, and satisfaction level.

UX goes beyond the user interface (UI) and encompasses the entire user journey, from the initial awareness and discovery of a product or service to the ongoing usage and post-interaction reflections. It involves understanding users' needs, goals, and preferences and designing solutions that meet those needs effectively and efficiently. [9, 10]

Questionnaire

A questionnaire is a research instrument commonly employed in usability testing and other fields for collecting structured, quantitative, and qualitative data from participants. It consists of a series of questions or prompts designed to gather information about a specific topic or user experience, often through self-reporting. Questionnaires can be administered on paper, digitally, or via online platforms, and they are often used to gather user feedback, measure satisfaction, or evaluate the usability of a product or system. By analyzing the responses to questionnaires, researchers can identify patterns, trends, and areas for improvement.

System Usability Scale

The System Usability Scale (SUS) is a widely used, standardized questionnaire that focuses on evaluating the usability of a product, system,

or service. Developed by John Brooke [11], the SUS comprises ten statements, each of which is rated by participants on a five-point Likert scale (from "strongly disagree" to "strongly agree"). The SUS provides a quick, reliable, and cost-effective means of assessing the perceived usability of a system.

2.2 Technical background

2.2.1 Programming for iOS

iOS is a mobile operating system developed by Apple Inc. for its suite of mobile devices, including the iPhone, iPad, and iPod touch. Building applications for iOS requires a unique set of tools, languages, and frameworks, which are designed to ensure seamless user experiences and optimal performance on Apple devices.

At the moment of writing this master thesis, there were two frameworks that could be used to write an iOS application:

1. UIKit [12] - UIKit is a framework for building UIs using Objective-C and Swift programming languages, and it has been the go-to choice for iOS developers since the introduction of the iPhone. It provides a comprehensive set of UI components and tools for designing, implementing, and managing user interfaces on Apple platforms.
2. SwiftUI [13] - SwiftUI is a more recent framework introduced by Apple at Worldwide Developers Conference (WWDC) in 2019. It is a declarative UI framework that allows developers to create user interfaces using Swift code. With SwiftUI, developers can describe their UI components and their states, and the framework takes care of the rest, automatically updating the UI when the state changes. SwiftUI is designed to work seamlessly across Apple platforms and is built with Swift, making it easier for developers to adopt and work with.

2.2.2 Embedded programming

Embedded programming is a specialized domain within software engineering that involves the design, development, and optimization of software for microcontrollers, microprocessors, and other embedded systems. These systems are prevalent in a wide range of applications, including consumer electronics, automotive systems, medical devices, and industrial automation. Embedded programming is characterized by its focus on resource-constrained environments, which require engineers to implement efficient algorithms and optimize software performance in terms of power consumption,

memory usage, and real-time response. As such, embedded programmers must have a deep understanding of low-level programming languages, hardware architectures, and system-on-chip (SoC) components. [14, 15]

2.2.3 Bluetooth protocol

Bluetooth is a wireless technology standard used for exchanging data over short distances, typically between personal mobile devices, laptops, controllers, etc. It was invented in 1994 by Ericsson, the Swedish telecommunications company, and is now an open standard maintained by the Bluetooth Special Interest Group (SIG).

Bluetooth devices communicate with each other using a master-slave model, where one device acts as the master and the others act as slaves. The master device manages the communication between all the other devices, and the slaves follow the master's instructions. This model is used to ensure that there is only one device transmitting data at any given time, which helps to avoid collisions and improve the reliability of the communication [16].

Bluetooth Classic & Bluetooth Low Energy

Bluetooth Classic and Bluetooth Low Energy (BLE) are both wireless communication technologies that use the same underlying Bluetooth technology. However, they are designed for different purposes and have several key differences.

One of the main differences between Bluetooth Classic and BLE is their power consumption [17]. Bluetooth Classic consumes more power than BLE, as it is designed to support high-bandwidth data transfer such as streaming audio or transferring large files. BLE, on the other hand, has been optimized for low power consumption and is ideal for devices that run on battery power and do not require much bandwidth such as fitness trackers, smart home devices, and beacons.

Selected Bluetooth classic profiles

Bluetooth Classic specification defines many different profiles for a variety of use cases. Some of the most common are:

- GAP (Generic Access Profile) - defines how two Bluetooth devices establish a connection,
- A2DP (Advanced Audio Distribution Profile) - used for streaming audio in high quality,

- AVRCP (Audio/Video Remote Control Profile) - responsible for media control like pausing and resuming,
- HFP (Hands-Free Profile) - allows to make basic actions like making calls through a peripheral, commonly used in e.g. cars,
- HID (Human Interface Device Profile) - provides support for devices like keyboards, joysticks etc.

GATT Client and Server

General Attribute Profile (GATT) [17, 18] is a protocol used in BLE in which a client device can read and write values to/from the server device and server device can send notifications to the client device. A device defines its capabilities via **profiles** which have following, hierarchical structure:

- **Services** - are the primary way of organizing and grouping related data and functionality on a BLE device. They represent a specific aspect of the device's functionality, such as heart rate monitoring, battery level reporting, or environmental sensing.
- **Characteristics** - are data elements that represent a specific piece of data or functionality within a service. Each characteristic can be interacted with in several ways, e.g, read or write. For example, the heart rate measurement characteristic of a heart rate monitoring service represents the current heart rate of the user.
- **Descriptors** provide additional information about a given characteristic. They are used to describe a characteristic's value in a human-readable way, specify how it should be displayed on a user interface, or provide additional metadata about the characteristic, such as its units of measurement. Descriptors are optional, and multiple descriptors can be associated with a single characteristic.

The Fig. 2.3 shows an example of how a hierarchical GATT structure could look like in an example device, namely a tabletop clock with built-in thermometer. Although only two services are presented (with one detailed), there are not any constraints of how many could be defined - the same applies to characteristics and descriptors.

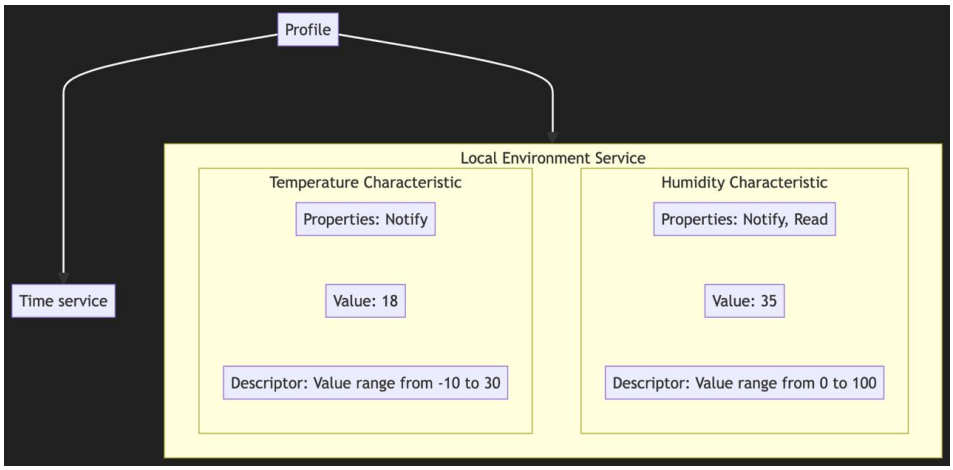


Figure 2.3: Example of how the GATT profile could look like.

General Attribute Profile Server/Client are roles a device can take - in general a server is the device that produces data and client consumes them. The two roles are not exclusive, though, and a device can be a client and server simultaneously, e.g. earbuds consuming audio data sent by phone but at the same time broadcasting data from built-in accelerometers. Also, a server device can have multiple client devices connected to it at the same time.

ANCS

Apple Notification Center Service [19] is a protocol used by Apple devices, such as the iPhone or Apple Watch, to provide notifications from iOS applications to external devices. ANCS allows third-party BLE accessories (like headphones, fitness trackers, smartwatches, etc.) to take the role of a GATT client and receive notifications from iOS devices, which is GATT server, without the need for a dedicated app or a continuous connection. Being a standardized protocol, it improves the interoperability between different devices and makes it easier for developers to implement communication between their accessories and Apple devices.

It is worth noting, that this is in fact the only way for accessories to receive notifications from Apple devices.

Chapter 3

Method and Prototypes

3.1 Defining core functionalities

The very first thing we had to do was to think of a suitable set of functionalities that Xocchiali AR glasses would have. After brainstorming sessions with Xocchiali team we came to a conclusion that he main premise behind the Xocchiali AR glasses is to notify the user when certain events occur, for instance when a text message is received. The user should be able to configure what happens depending on what event occurs. They should be able to differentiate between messages from different people or notifications from different applications. Moreover, since the glasses have an input mechanism in form a button, it can be used to trigger user-configured actions, for example calling relative upon clicking the button once or setting a timer upon clicking twice. Having defined that, we derive from it that the application should have following functionalities:

- Onboarding process - a kind of interactive welcome screen that would guide the user in first-time use of the glasses. It would show the user how to connect the glasses and explain what functionalities are there in an accessible manner. Note that this was not a main point of focus as we did not have access to the glasses themselves (see Section 1.4 Limitations).
- Setting up **alerts** - this functionality should allow to configure how the glasses react upon receiving a notification - depending on the person or source (application which sent the notification) the glasses should blink in a desired color and/or produce a sound and mobile app should allow to personalize that.
- Setting up **triggers** - triggers are actions that should happen when a user performs a predefined interaction with the glasses, namely: single

button tap, double button tap and triple button tap.

- Settings screen - there a user should find any other necessary information about the app or the glasses (e.g. serial number). It should help the user in case they need it, e.g. by providing redirection to frequently asked questions or providing contact to support.

Having these functionalities defined early on was crucial for choosing appropriate microcontroller to be used in the glasses, as it is a one-time decision that requires ordering specific hardware and it cannot be easily changed later during the project.

The work on this thesis consisted of two parallel, rather independent paths:

- Mobile application - when it comes to the mobile application (working title **Xocchiali Manager**) we followed agile[20], user-centered and iterative process.
- Qualcomm microprocessor - for the board we took a different approach, namely waterfall approach [21]. There are two main premises behind that:
  - Due to the nature of hardware being not as easily modifiable as design or software.
  - Due to the fact that it deals with strictly technical parts of the product that are hidden from the end users.

Since both parts follow different methodology and are fairly independent, firstly we will focus on describing how the development process of the mobile application looked and then we will describe the work on the Qualcomm microprocessor.

3.2 Low fidelity (Lo-Fi) prototype

Having defined core functionalities of the application, we proceeded to create the first, low fidelity prototype. As it is very cheap to make and can easily pinpoint flaws in the product design, content or interaction, it is a very natural step in an iterative design process.

We decided to create paper prototypes as they are very fast and easy to make and they are not constrained by any technical limitations. First, we agreed to each develop our own paper prototype so that we do not influence each other in the process. In result we had two following storyboards.

3.2.1 Adam's prototype

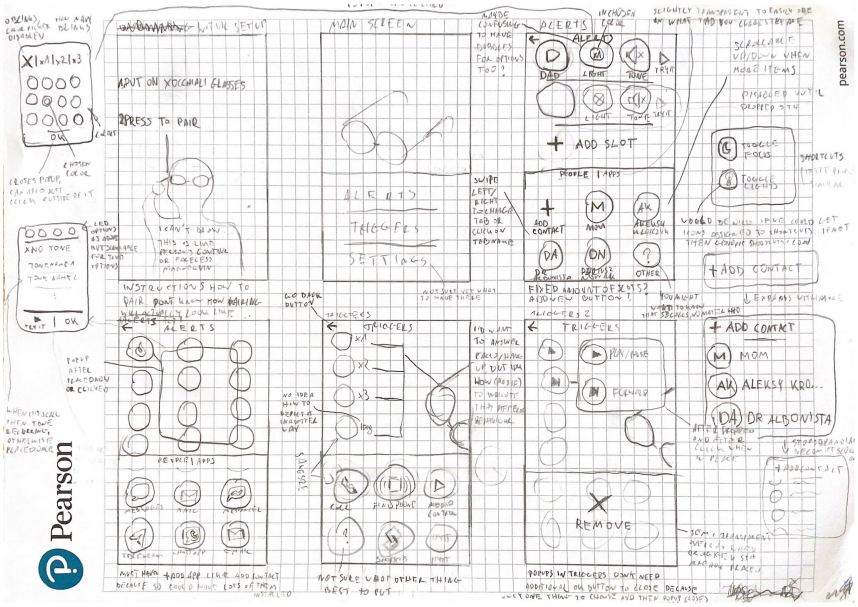


Figure 3.1: Adam's low fidelity prototype

Source: own

Before we made our prototypes, we had been shown some really simple mockups of the application by Xocchiali. It was just a plain main menu screen with glasses in the spotlight and three buttons below in one and triggers screen in the second with glasses more to the side, three trigger slots and six bubbles to choose from below. It gave a little bit of understanding what Xocchiali might have in mind, but it was not much more than that. It gave an idea that the connected model of the glasses should be displayed indicating status of connection and acting as an elegant visual hint to what triggers relate to and what kind of design is desired. This prototype is an attempt to take these initial ideas, fill all the missing pieces and design the whole interaction around it.

When the user turns the application for the first time and thus does not have their glasses connected yet, it would display instructions both in text and visual what to do to get the glasses to pair. Once connected, the user would be displayed a simple, yet elegant main menu from where they could choose which screen to navigate to alerts, triggers or settings. The idea was to keep as low a number of screens as possible and just these should be enough if context actions are shown in simple popups. Settings view was

not included in the prototype as the interaction with it was deemed trivial and not that important at that point.

There are two alternative designs for the alerts screen included. Both of them were split in two parts - upper one where one can see alerts that one has already configured and lower where one has alerts to choose from. The lower part was exactly the same for both. It had two tabs, one for people and one for apps and in each there is a scrollable view with bubbles to choose. The difference between these alternatives was that the first one had a scrollable list of alerts, which one could have as much as one would like and could instantly see how exactly would the system react to a given notification, while the other has predefined grid of slots where one can see all notifications triggering alert at once but with less immediate details.

Triggers screen has similar, consistent design with alerts screen. In the upper part there are slots for the respective triggers and in the lower part there are actions to choose from, also in the form of the bubbles.

Interaction in both alerts and triggers screen was mostly the same. A drag and drop gesture is used on bubbles to move them to slots (or to remove them). While configuring alerts the same kind of popup would appear no matter what notification is chosen. However, in triggers every action has its own popup. Notification bubbles in alerts disappear when moving them to an empty slot - it is not possible to define two different alerts for the same notification. In contrast, in triggers screen possible actions do not disappear or become disabled once in use - this is because one can configure a given action in different instances to do something else. Additionally, even if configured in such a way that two different triggers result in the exact same action - there is no issue with that.

3.2.2 Aleksy's prototype

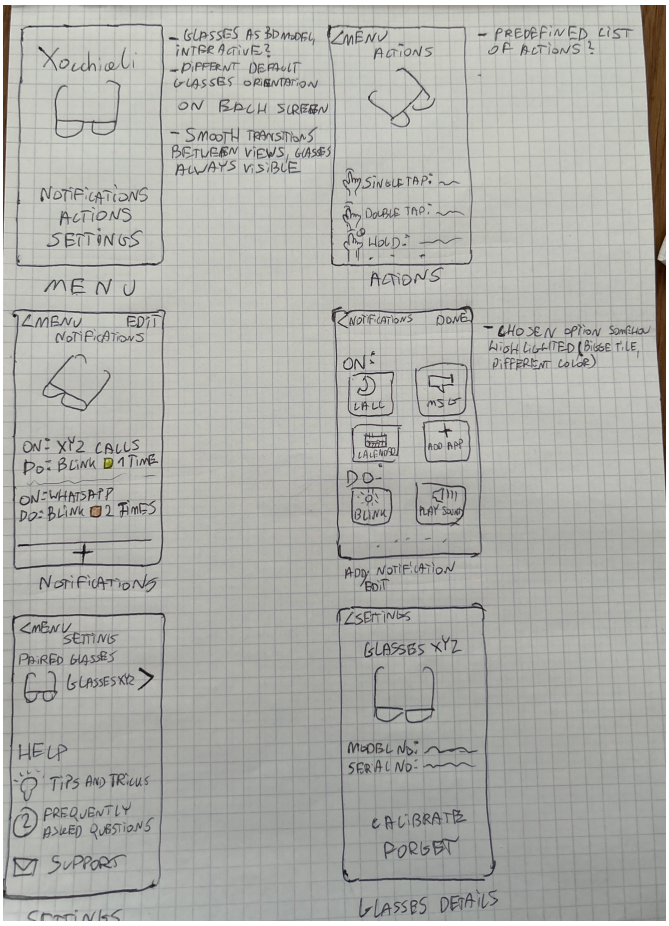


Figure 3.2: Aleksy's low fidelity prototype

Source: own

This prototype was inspired by system iOS applications like settings, shortcuts or clock as well as Apple's Human Interface Guidelines [22]. The prominent part are the glasses that are visible on every screen that change their orientation via an animation - the idea is that they match the user's physical model (assuming Xocchiali would have more than one model of glasses). The main menu is simple showing company's name and three navigation buttons leading to consecutively: **alerts** screen (named "notifications" in the prototype), **triggers** screen (named "actions" in the prototype) and **settings**. Every screen contains a title header that informs the user where he is at the moment as well as necessary context actions on the top of the

screen, e.g. to go back to the main menu.

The **alerts** screen consists of a predefined list of actions that should happen when the user clicks the button a specific number of times - user should be able to tap them and select desired outcome in a context screen that would appear.

The **triggers** screen contains another list - this time it is scrollable and shows user-defined alerts. Every entry in the list shows what happens when a specific event happens, e.h. when a specific contact calls. On the bottom there is a plus button that allows users to add new entries to the list. Upon tapping it, a context screen that would guide the user in personalizing a new alert.

The last view is a settings view that provides the user with information about the user's physical, connected glasses as well as references to tips and tricks, frequently asked questions and support.

The idea behind this prototype was to make each screen relatively simple and break complexity into pieces by hiding it in context view so that the screen never contains too much information.

3.2.3 Prototypes testing

Both prototypes were presented to four users in the exploratory testing session to decide which design choices are good and which still need refinement or be completely redone. Two of the test users were Xocchiali representatives, one was our supervisor and the last - a friend of ours. Their age spanned approximately from 20 to 50 and all of them but our friend were male. It was conducted in form where we described our ideas and sketches about different screens and then discussed them with the users. What we found out is that in general the circular design from Adam's prototype appealed to the users more than e.g. simple list as it felt natural and made the app more original and standing out. Additionally, the drag and drop gestures felt intuitive so we decided to continue with this option rather than having the lists - another advantage of that was that interfaces that are less iOS-like would be more suitable for devices running Android operating system as the final application is aimed to be multiplatform. Moreover, this design was a bit more aligned with what Xocchiali had in mind and so they were naturally leaning towards it. When it comes to the settings screen, we found that what Aleksy had proposed is clear and easy to understand and so we chose to develop it further in the future. We also had an internal brainstorming session where we redesigned popups in the alerts screen because we concluded that initial designs were not good enough - they tried

to make alerts configuration components very compact while there was no real reason for it.

To sum up, both prototypes had their advantages and disadvantages and what we decided to do was to take what was good and combine it later during the hi-fi prototype phase.

3.3 High Fidelity (Hi-Fi) Prototype

Knowing advantages and disadvantages of our lo-fi prototypes, we began working on a high fidelity version. A few approaches were discussed, e.g. to leverage design-oriented applications like Figma but in the end we decided to do it as a mobile (iOS) application from the very beginning. We had planned to start with just the views without any functionality and introduce them further during next iterations. This decision was made for the following reasons:

- Both of us have experience developing applications for iOS but neither of us have great experience using a tool like Figma - we concluded that by developing views for the application we can do it relatively fast and produce a hi-fi prototype of higher quality.
- Done this way, the prototype would be a better visualization of the final, finished application.
- Created views could be reused later without having to implement them from scratch.

3.3.1 Note about our work manner

Before we present the outcome, we would like to describe how our work process looked like. Since we decided to make an iOS application, we had to choose between UIKit and SwiftUI framework and we went with the latter as it is meant to be a successor of older UIKit and it is, by our preference, more pleasant to work with. To make an application using this framework (the same would apply for UIKit), one has to use Apple's Xcode integrated development environment [23]. The very nice thing about Xcode is that it presents the developer with a preview of the user interface, allowing for very quick changes without the need to recompile the code every time which makes it a surprisingly good tool to prototype. Additionally, some UI changes can be made without even having to write code making it a semi "what you see is what you get" editor.

We used the Git [24] repository hosted on GitHub [25] as a version control

system [24, 26]. This allowed us to easily work on the project simultaneously, and ensured that we do not lose much progress in case, e.g. a computer would break. Additional benefit is that we get a whole history of changes so we can always go back to any point of development in case we needed that.

To organize our work and to better visualize our progress we used Trello [27]. It is a project management tool that leverages a kanban board which is an implementation of kanban methodology [28, 29].

The application was tested on iPhone 14 in the beginning and later (see Section 3.3.3) also on iPhone SE (2nd generation) which have vastly different screen sizes and aspect ratios - 6.1 inches with aspect ratio of 19.5:9 and 4.7 inches with 16:9 respectively [30, 31]. It was done to ensure that the user interface looks appropriately regardless of the screen size.

3.3.2 First hi-fi prototype

In the first phase, our goal was to combine our lo-fi prototypes and create the four views for menu, alerts, triggers and settings without any functionalities. At the same time we wanted to lay the groundwork for the application's look, style and further iterations. We kept the application minimalistic and, according to what we thought, simple. Since we did not focus on the onboarding process the application assumed that the glasses had been already connected and showed the main menu.

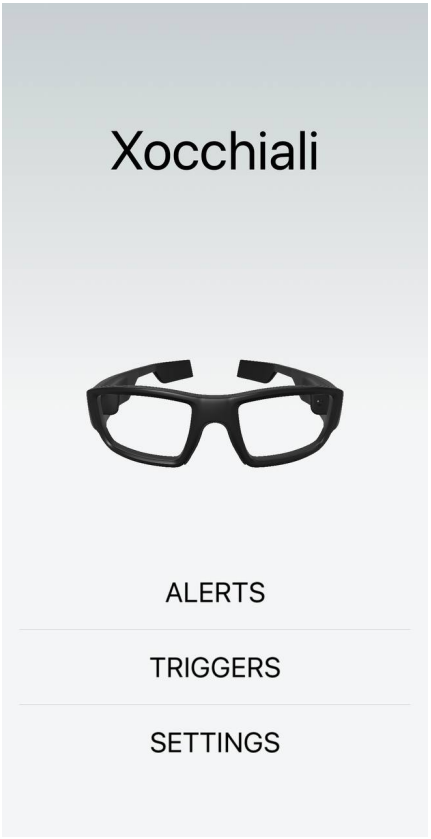


Figure 3.3: First hi-fi prototype: Main menu view

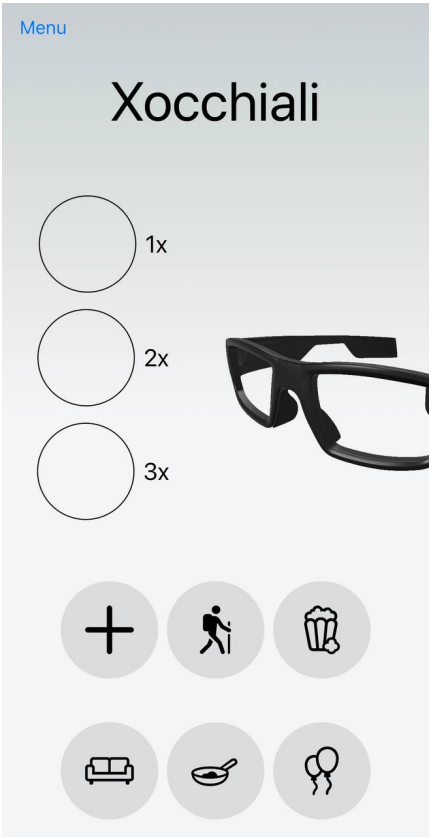


Figure 3.4: First hi-fi prototype: Triggers view

Source: own

The first thing the user would see is a main screen with a 3D model of the glasses [1]. The main menu screen also contains a title and just three buttons - we tried to keep the number of things on the screen as low as possible to avoid the feeling of confusion or being overwhelmed. We wanted the application to stand out so we made it as if it was "one screen", that is going to, e.g. triggers screen launched a smooth animation where the glasses scale and rotate while the UI elements are moved to either left or right edge of the screen.

The triggers screen contained three empty fields that depicted what should happen when the button on the glasses is clicked 1-3 times. Below a list

[1]Since we did not receive the actual Xocchiali glasses 3D model we used a model that we found on the Internet - all in compliance with the relevant license.

of actions can be found - in the beginning we used semi-random icons that did not represent anything specific as it was not important at that moment - we just focused on the fact whether it was understandable for the users to use drag and drop gesture in order to fill the empty spaces.

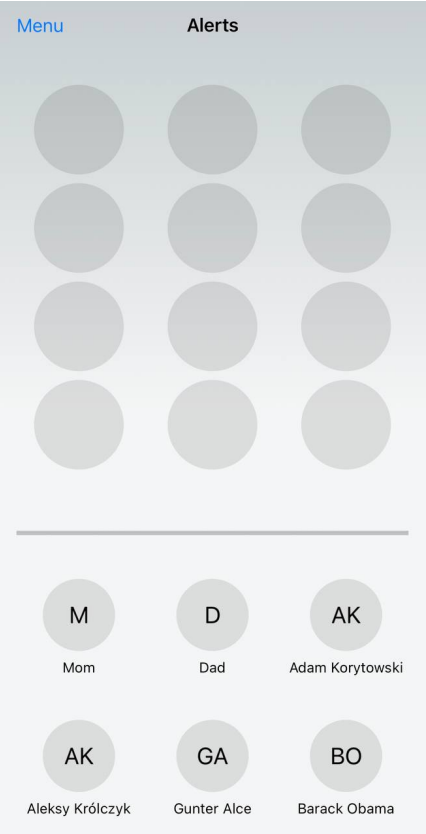


Figure 3.5: First hi-fi prototype: Alerts view

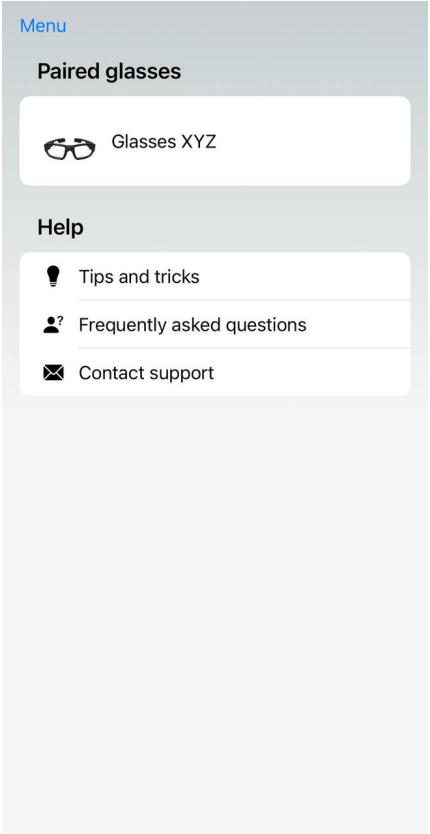


Figure 3.6: First hi-fi prototype: Settings view

Source: own

In the alerts view we could see twelve empty circles that depicted alerts already set by the users - all empty by default. In the lower parts of the screen there is a predefined list of contacts that the user could drag and drop above to set an alert, meaning that the glasses would somehow inform the user upon a message from the given contact. We thought that the screen looked somewhat intimidating and may be hard to understand but we handled that in the subsequent iterations using **help screens** about which we will talk about in a later (Section 3.3.3 Help screens).

The last screen is settings screen which shows the model of the connected glasses as well as can navigate a user to help in the form of tips and tricks, FAQ or support contact.

Testing and validation

After completing the first hi-fi prototype we tested it on 5 users including Xocchiali, supervisors and our friends. These tests were not very formal, were not recorded but we took notes from our observations and test subjects' comments. Combining this data with our own thoughts and ideas after more brainstorming sessions, we came to the following conclusions:

1. The users liked the minimalistic design of the application as well as they appreciated the animated transitions very much.
2. The users liked how simple the application is and that it consists of a very small number of different screens and functionalities.
3. Some of the users had trouble understanding the drag and drop gesture. The issue was that in order to start this gesture, a given interface element that was to be dragged had to be held for approximately 0.4 seconds. After this time, the element "popped out" of the screen and started following the user's finger - this is where some of the users had their "Aha! moment" [32] and from this point onward did not struggle with this gesture. However, as it turns out, this time interval was big enough for the users to think that the UI element is not interactive and they stopped the gesture before it even started.
4. Regardless of whether the users had trouble with the drag and drop gesture or not, they appreciated that the two main screens, triggers and alerts, were very similar and followed the same principle.
5. The application lacked a detailed view containing information about paired glasses.

3.3.3 Final hi-fi prototype

Being more knowledgeable after the tests and having more conversations about possible improvements, we began to enhance our high fidelity prototype. During the next two months or so we were working on improving the design and usability, as well as adding some new functionalities. Some of the newly introduced things were adding new features (e.g. alerts configuration) whereas others were trying to resolve issues we had found during test sessions with the first version of low-fi prototype. Last thing that we worked

on was making the design of the application more consistent, for instance ensuring that the font sizes are appropriate across different screens.

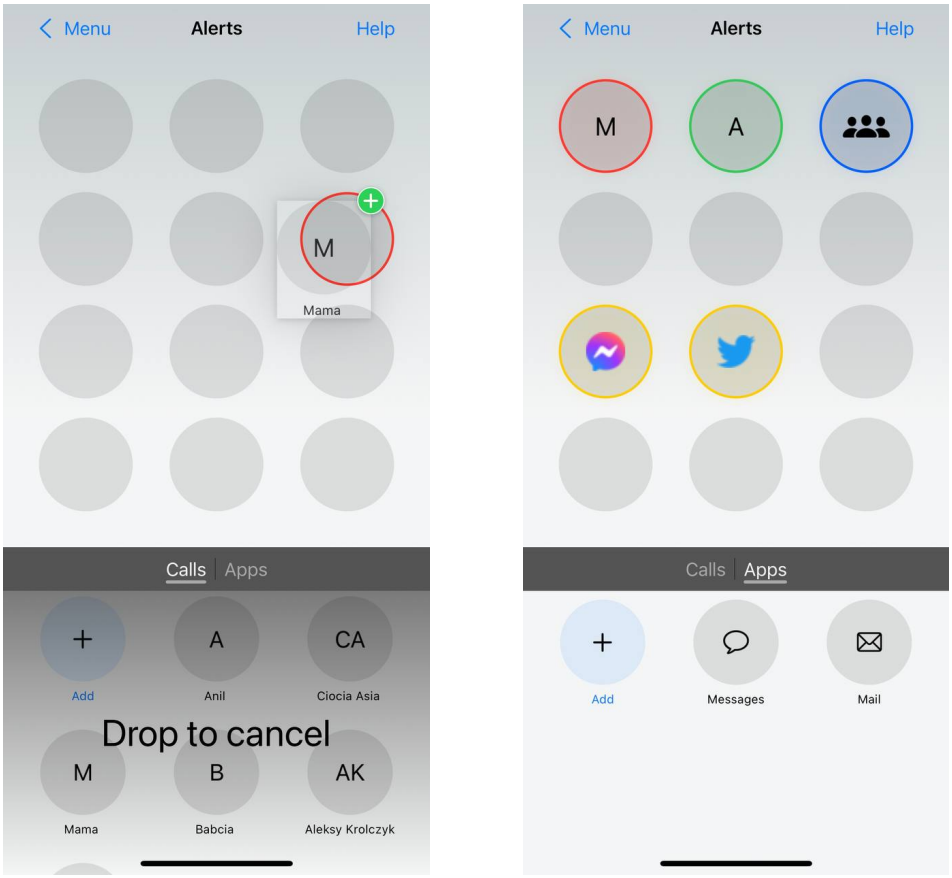


Figure 3.7: Final hi-fi prototype: Alerts views

The first image shows the moment of dragging a contact to be personalized. The second one shows a complete example setup - the color of the border/background shows what color is set for a given alert.

Source: own

Compared to the first version of our hi-fi prototype, at first glance the main difference is the lower part of the screen. Instead of a predefined list of contacts now the view is tabbed allowing to choose between contacts and applications. By default no contacts are shown but they can be added from a list using a plus button - the contacts are synchronized with the actual user's contacts book. Similarly, for the **Apps** tab, only two system applications

are shown, Messages and Mails, while any other applications have to be added in order to be used. After dropping contact/application onto the empty space in the grid above, a configuration screen (Fig. 3.8) appears and lets users define what should happen when receiving a notification - here a user can choose the color and number of blinks as well as personalize sound notification.

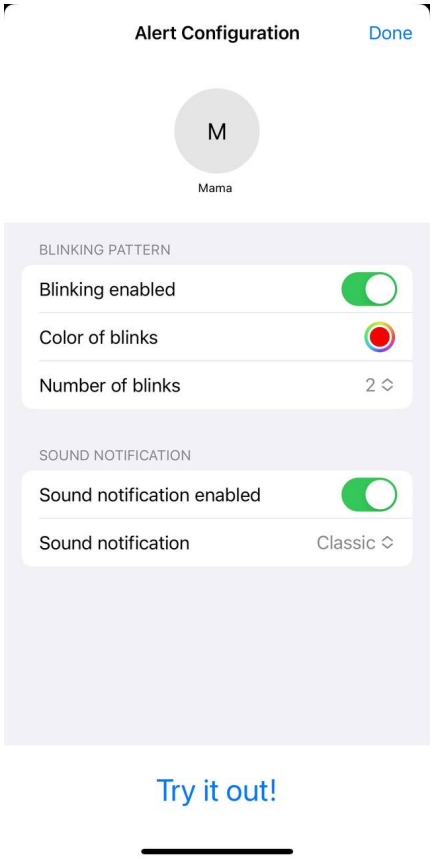


Figure 3.8: Final hi-fi prototype: Alert configuration view

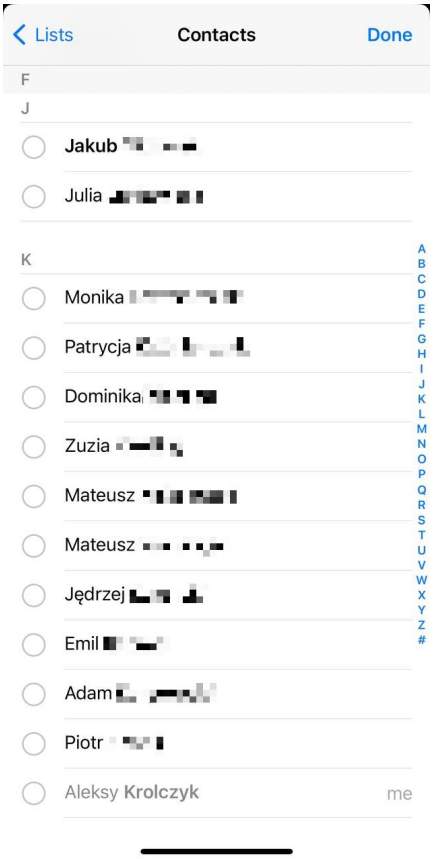


Figure 3.9: Final hi-fi prototype: Contact picker.

The names last names in the second image are pixelated due to privacy concerns. Source: own

In the triggers view (Fig. 3.10), the biggest difference is also in the lower part of the screen - placeholder actions were replaced with actions that could be used in a final application, like setting a timer or calling a contact. The circles also were extended by a title caption and resized so they better fit the application. Each of the triggers had its own context menu, e.g. upon

dragging and dropping "Call" trigger a menu with contacts appeared to select a particular person. Additionally, the labels next to the circles in the upper part of the screen received an icon and selected action title so that it is easier to understand what they represent.

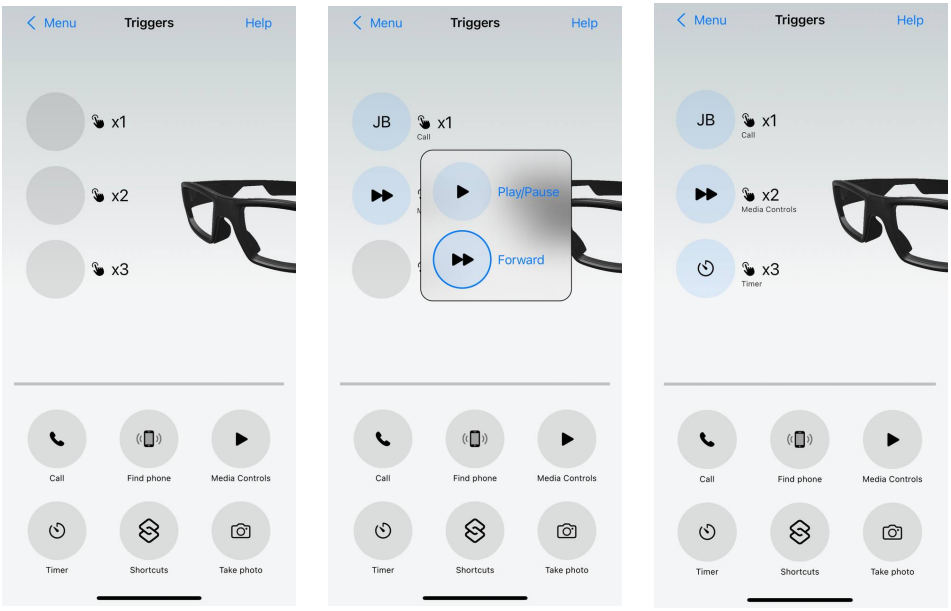


Figure 3.10: The first image shows the triggers when nothing is configured yet. The second one shows a context menu for "Media controls" trigger. The third one depicts a complete, example configuration.

Source: own

The settings view (Fig. 3.11) did not change very much; it is not the central part of the companion application. Nevertheless, we added **Manage** section which allowed the end users to add and remove contacts or applications that had been added in either alerts or triggers view. While the first of the actions may not seem very helpful as it can be done from the views themselves, the removal action is not possible there so in case somebody e.g. added too many contacts, they can deal with that here. The second change is adding navigation to the glasses details view (Fig. 3.12) which displays a model number, serial number and allows for disconnecting glasses. Possibly more detailed information could be found there in the future.

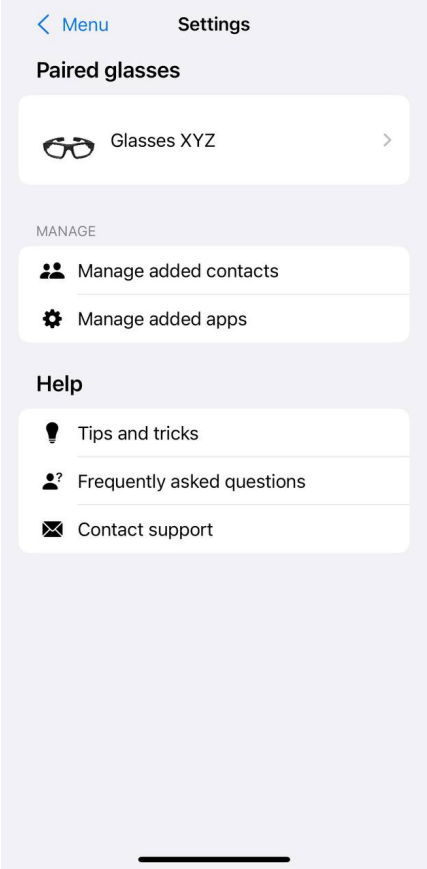


Figure 3.11: Final hi-fi prototype: Settings view

Figure 3.12: Final hi-fi prototype: Glasses details views

Source: own

Help screens

Despite many improvements that had been developed over the weeks, the tests strongly suggested that the minimalistic look of the application may be obscure and daunting for the new users. While some of the instructions might be conveyed during the onboarding process, there still should be some kind of help in case the user gets lost.

To alleviate that problem we introduced **help screens** for both alerts and triggers views. These screens would appear automatically when a user first time enters either alerts or triggers view but could always be invoked later using the **Help** button found in the upper right corner of the screen.

The help sequence consists of an arbitrary number of child screens (in our case it was three) that explain what and how can be done in a given section of the application, e.g. in alerts view.

In order to make these screens help the users understand how the application work, we used two different techniques:

- Explanatory texts.
- Animations that show how to interact with some of the elements in a given view.

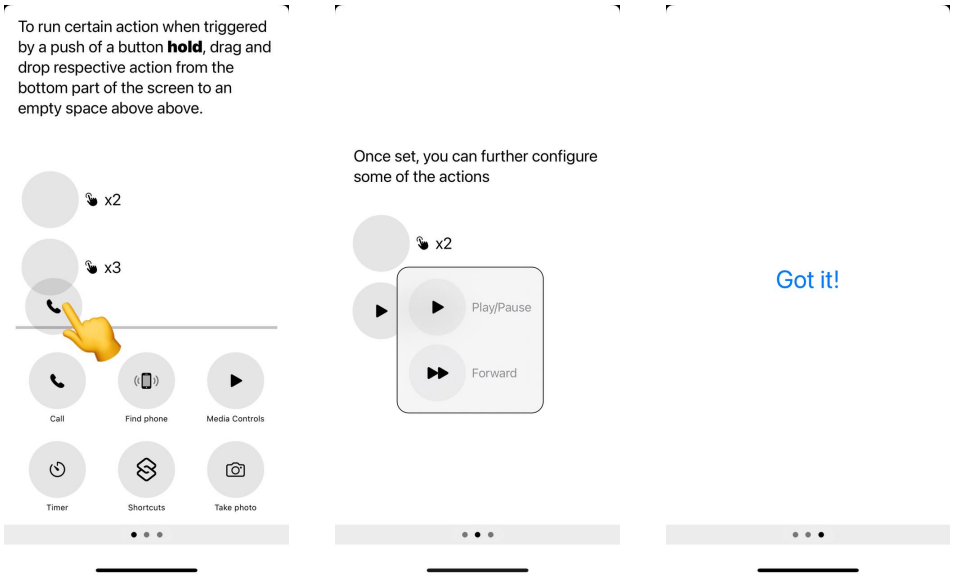


Figure 3.13: Final hi-fi prototype: Triggers view help screens

Source: own

More device support

In the beginning we tested the application only on iPhone 14. That resulted in unconsciously making some of the screens assuming a few device-specific characteristics. When we began testing on iPhone SE (2nd generation) this became apparent immediately. When we realised that, we started adding support for different resolutions and aspect ratios and made sure that it would work seamlessly on both our testing devices from then onward. Even though we did our internal testing only on these two physical devices, we further verified that the UI looks appropriately using virtual devices with different characteristics.

3.4 Glasses microcontroller application - proof of concept

As another part of the thesis, we also explored the feasibility of adapting the Qualcomm microcontroller, which is designed with Bluetooth audio devices in mind, for the Xocchiali glasses. The result of that was a proof of concept application that can be used as the foundation for the actual product.

3.4.1 Microcontroller

At first we were provided with the Qualcomm QCC3071 microcontroller [33]. However, it turned out to be designed for earbud type devices, whereas the glasses would work like a headset. The main difference is that a microcontroller designed for an earbud device supports only a single audio channel as it is supposed to be paired with a second device of the same type for stereo audio. There is no need for that in the glasses as they form a single object already and it would only add more cost to production, consume more power as well as make the communication protocol more complicated. After determining that it is not a right microcontroller, we received a different one, namely the Qualcomm QCC5171 [34]. It has a very similar set of capabilities to QCC3071 but also supports stereo audio and can be used for a headset application.

Qualcomm provides everything one may need to start developing a headset application for their microcontroller: an IDE, debugging tools, drivers, a wide set of libraries and an example project. That example project is something that drastically speeds up development as it makes, without any modifications, already a working example of a Bluetooth headset which you can pair with, play music, control it with buttons, answer calls etc. Another benefit to it is that it shows how to properly use modules and libraries. It means that one does not really have to deal with all the things that are expected from every Bluetooth headset anyway and can just focus on developing additional modules that are specific to a given product.

What is worth noting is that it is required to sign a non-disclosure agreement with Qualcomm to be allowed to develop anything for this family of microcontrollers and therefore we are limited in what we can write in this section.

3.4.2 Functionalities to develop

As was described previously Section 3.1, we decided on a few core non-standard for ordinary Bluetooth headset functionalities that would for sure be required in the product. That is handling configurable alerts and trigger-

ing configurable actions with the button clicks. The designed architecture of these functionalities can be seen on Fig. 3.14.

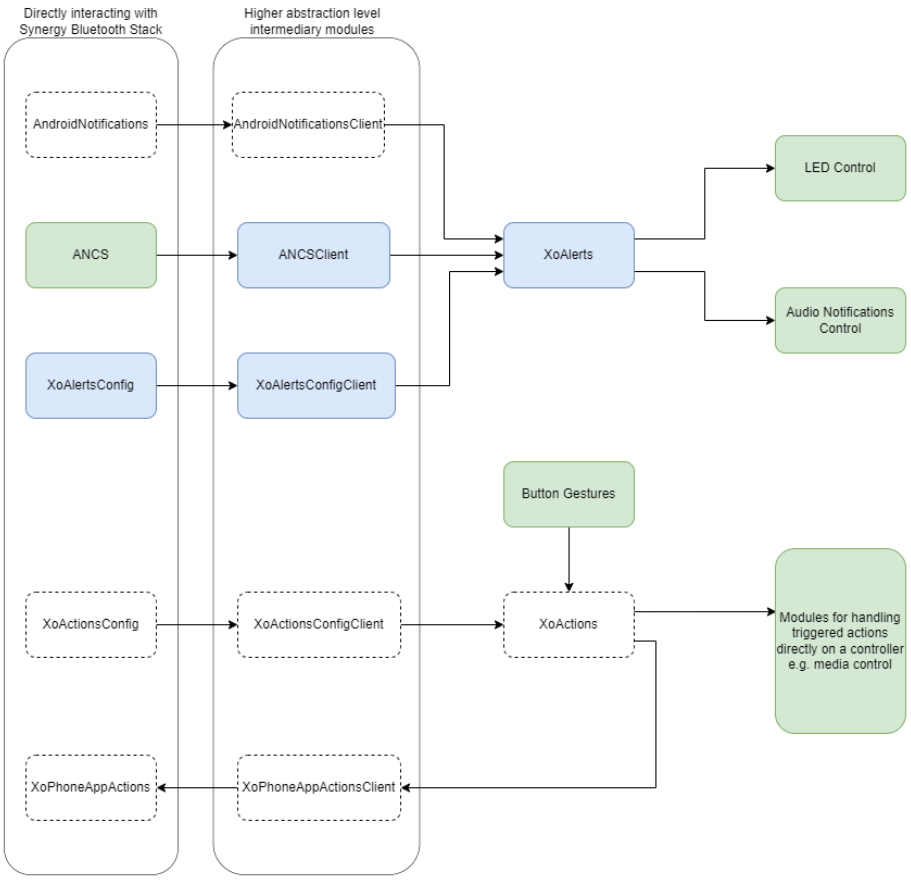


Figure 3.14: Architecture of the part of the system that is responsible for Xocchiali's alerts and actions features. Green blocks - modules already provided by Qualcomm, blue blocks - modules implemented by the authors, blocks with dashed borders - modules not implemented yet. Arrows represent flow of the messages between modules.

Source: own

Modules in the architecture:

- Alerts
  - AndroidNotifications - a module that implements communication with other devices over Bluetooth GATT client which is notified

when a paired Android device receives a notification such as text message, email etc.

- AndroidNotificationsClient - module that puts AndroidNotifications module to use and provides higher abstraction to it: initializing it, managing its state, managing data read from/written to it.
- ANCS - Apple Notification Center Service, module that implements communication with other devices over Bluetooth GATT client which is notified when a paired Apple device receives a notification such as text message, email etc.
- ANCSClient - a module that puts an ANCS module to use and provides higher abstraction to it: initializing it, managing its state, managing data read from/written to it.
- XoAlertsConfig - a module that implements communication over Bluetooth GATT Server from which you can read and write to configuration of alerts.
- XoAlertsConfigClient - a module that puts XoAlertsConfig module to use and provides higher abstraction to it: initializing it, managing its state, managing data read from/written to it.
- XoAlerts - module that reacts to notifications from a paired device in a way that is specified by a given configuration e.g. by triggering blinking an LED or a playback of a sound.
- LED Control - module responsible for playing a specified LED blinking pattern when told to.
- Audio Notifications Control - module responsible for playing a specified audio file when told to.

- Triggers
  - XoActionsConfig - a module that implements communication over Bluetooth GATT Server from which you can read and write to configuration of actions.
  - XoActionsConfigClient - a module that puts XoActionsConfig module to use and provides higher abstraction to it: initializing it, managing its state, managing data read from/written to it.
  - Button Gestures - a module that is responsible for recognizing button press patterns e.g. single click, double click etc.

- XoActions - a module that reacts to triggers in a way that is specified by a given configuration e.g. by triggering pausing a song or calling somebody.
- XoPhoneAppActions - a module that implements communication over Bluetooth GATT Server which notifies the paired device that it should run a specific action e.g. making a HTTP request, running Apple Shortcut, starting a timer etc.
- XoPhoneAppActionsClient - a module that puts XoPhoneAppActions module to use and provides higher abstraction to it: initializing it, managing its state, managing data read from/written to it.
- a set of modules that can be used to handle actions that do not need to be triggered directly on a paired device because there is just no need for it or because there already are designed Bluetooth protocols for it e.g. media control via AVRCP (Audio/Video Remote Control Profile) or starting a call via HFP (Hands-Free Profile).

3.4.3 Implementation

Similarly to the development of the mobile application, we created a Git repository hosted on GitHub as a version control system for the codebase. After familiarizing ourselves with the development environment, initial setup of the development board and some configuration of the project, we successfully ran the application on a microcontroller for the first time. Then we thoroughly tested that all basic Bluetooth headphones functionality works as expected, which includes such things as pairing with the mobile phone, playing music streamed from a phone, controlling media playback with the buttons etc. We tweaked the configuration to e.g. disable unwanted functionality like Swift Pair[35] as it was not needed for our application. At that point we were ready to implement our custom functionalities.

The first thing we decided to do is to set up an Apple Notification Center Service (ANCS) client. It is a BLE GATT service where an iOS device as GATT server which in short notifies clients about notifications arrival, modification, deletion and from which clients can read requested details of notifications or applications. ANCS is the only way one can get notifications from an iOS device. That means that basically the whole alerts functionality is based on it when it comes to iOS devices and therefore it is crucial to have it operating. After

some work we managed to develop the module and include it in the app that successfully retrieves all of the needed information when a paired iOS device receives a notification.

What came next was to develop a custom GATT service which would reflect alerts settings you could make on a mobile app. The microcontroller would take the role of a server which the mobile app would write the configuration to and read it from it, making it the source of truth for this configuration. For every alert configuration we needed following attributes:

- application or contact identifier (depending on alert type),
- number of LED blinks,
- LED blink color,
- tone signal.

We settled on a fixed number of alerts that could be configured, therefore we thought that it is appropriate to have each of the attributes above for every possible alert slot a separate characteristic within a GATT service. This means we had $12 \cdot 4 = 48$ (number of alert slots $\cdot$ number of attributes) characteristics in our service. As one can see in the Fig. 3.8 there is also a setting about blinking and notification by sound being enabled. There is no need to have these values represented separately here as number of LED blinks set to *0* and tone signal set to an enumeration value of *none* would translate to respective switches in the mobile application being set to a toggled off state.

Once we had information about the user's configuration of the alerts and notifications from a paired iOS device available in the application run-time we could combine those to run desired audio-visual signals. Compared to the previous steps, this one was not only less complex but also we had much better understanding of the system by then and therefore developing this functionality went very smoothly with no major issues.

At that point we concluded the work with the application for the microcontroller as we were running short on time and we wanted to focus on the application and related usability study. Nonetheless, we had a functioning proof of concept for the alerts functionality. What was still missing out of the major functionalities at that point was the module to receive notifications from Android devices and some of the features related to triggers.

3.5 Usability Study

Having completed the hi-fi prototype, we wanted to finally evaluate its usability on a group of people. The following sections contain a detailed plan procedure we followed during our usability study, whereas the results, including e.g. information about the test subjects, are presented in the next chapter (Chapter 4 Results and Discussion).

In the beginning we planned to do tests solely in English but it turned out that some of our test subjects were Polish and did not know English well enough. Since we are Polish as well, we prepared two versions of the tests for both languages - each version with translated briefing/debriefing, questionnaires (including proposed Polish translation of the Standard Usability Scale[36]) and instructions.

When preparing the usability study, we followed the guidelines of Rubin and Chisnell [8]. In particular, we prepared our test plan using the following structure:

- Purpose
- Research Questions
- Selection of subjects
- What data to collect
- Task list
- Procedure (test session design)
- Test environment/equipment
- Test moderator role, and the role of others

Purpose

The purpose of this study was to measure the usability of the mobile application developed for this thesis. The application should guide the users and allow them to configure the necessary settings to make their glasses usable and practical.

Research questions

To enhance the organization of our work, we focused on the following three research questions that we intended to answer upon completing the tests:

1. Is the mobile application usable?
2. Is the mobile application self-explanatory and successfully guides the user?
3. Is it easy for the users to configure the necessary settings?

Selection of subjects

We planned the test subjects to be users who might have been interested in smart glasses. They should have had basic knowledge of how to use a smartphone and be able to read and understand English or Polish. Wearing glasses was not a criterion that would disqualify potential test subjects.

What data to collect

Throughout the testing process we gathered both qualitative and quantitative data which included:

- Task completion or failure
- Time taken to complete the tasks
- Interventions of the moderator
- Key parts of the user interface/test that were troublesome for the test subjects
- Test subjects' comments
- Other moderators' observations and comments about how the test subjects coped with different parts of the test
- Answers from the introductory and concluding questionnaires

Task list

We produced a task list in order to systematically test different parts of the application. It consisted of the following parts:

1. Configuration of alerts
2. Configuration of triggers
3. Miscellaneous activities

The full table is available in the Appendix A.

Procedure

The procedure was standard and consisted of briefing, the actual test and debriefing all of which are described in the following sections.

Briefing

To ensure consistency in our briefing, we devised a script that was read to all participants. This approach aimed to eliminate any potential bias resulting from conveying slightly different information to each individual. During the briefing, we provided an overview of the study's purpose and its significance. We followed this with a brief explanation of Xocchiali's idea for the product and its capabilities, along with details regarding the test procedures. Lastly, we informed participants that the session would be recorded (audio and screen recording), emphasized that participation was entirely voluntary, and assured them of their right to withdraw at any time.

The whole briefing script in both English and Polish is available in Appendix B.

Upon obtaining participants' consent, we proceeded to gather some basic information, such as gender, age, and prior experience with similar technologies. The complete questionnaire in both English and Polish can be found in Appendix D.

Actual test

The test subjects were given instructions and an iPhone with our application installed and opened in the main menu. Since this moment, we started screen recording in order to, among others, measure the tests' length.

Whole instructions that were given to the user in both English and Polish are available in Appendix C

Debriefing

After the user finished the tasks we thanked them and proceeded to a SUS questionnaire. After filling it we conducted a semi-structured interview with the user to gather additional insights we might have missed otherwise. Questionnaire and questions we prepared for the interview, both in English and Polish are available in Appendix E.

Test environment and equipment

The test could be conducted in an arbitrary place as long as it was not distracting for the test subjects (e.g. in a noisy environment). As for the equipment, we gave the test subjects our own mobile phones. Apart from that, we used our computers during the tests - one to display the instructions and SUS questionnaire after the test while the other was used to gather notes and read prepared questions.

Test moderators roles

We planned to have two moderators during the tests.

- **Moderator 1** - Responsible for direct interaction with a test user. Conducted briefing before the test, during the test was a person that would answer any questions from the user and after the test would conduct debriefing.
- **Moderator 2** - Responsible for things that do not include direct interaction with a test user. Made sure everything was recorded, kept track of the time, and took notes.

Chapter 4

Results and Discussion

4.1 Usability study results

4.1.1 Quantitative data

In the end, we managed to conduct $n = 10$ test sessions. Six of the test subjects were males and four - females. While most of our participants could be described as young adults, we also conducted tests with older people - more detailed distribution can be found in Fig. 4.1. For the sake of the following chapter, we labeled each of the participants with consecutive letters of the English alphabet.

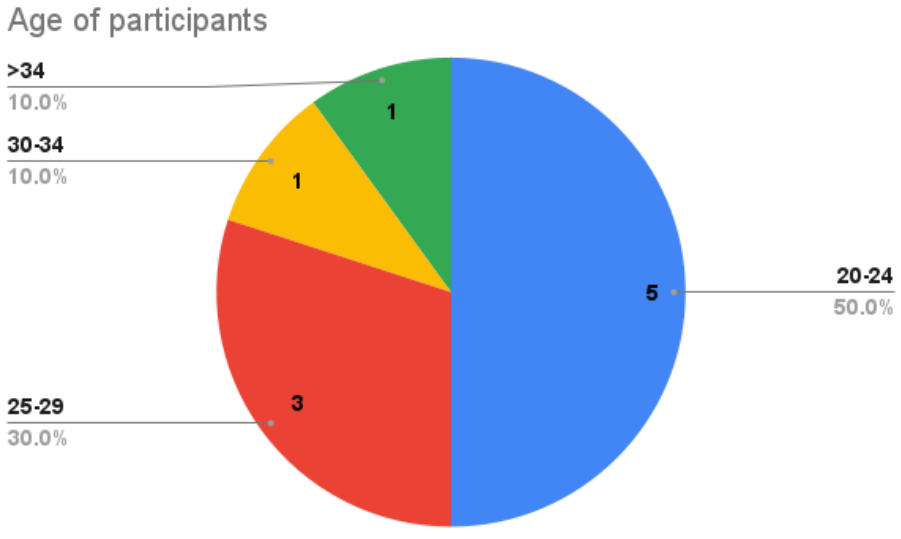


Figure 4.1: Age distribution

The tasks completion time for task 1 and task 2 had fairly big standard deviation but clearly task 2 was completed faster. While the expected completion time for task 1 was slightly longer than for task 2, we would argue that the results were further affected by the effect of transfer learning [8] as the instructions were always given in the same order.

The task 3 was expected to be completed relatively fast and here all of the participants met our expectations.

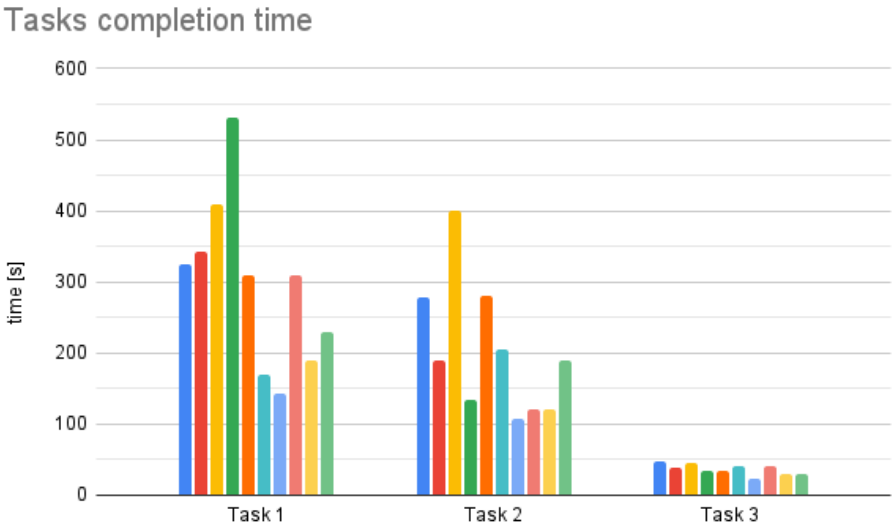


Figure 4.2: Tasks completion time

Table 4.1: Task success matrix

User	A	B	C	D	E	F	G	H	I	J
Task 1	×	×			×	×	×	×	×	×
Task 2	×	×		×	×	×	×	×	×	×
Task 3	×	×	×	×	×	×	×	×	×	×

Successful task completions were marked with "×" while the empty space means the task was not completed successfully.

Most of the users completed all of the tasks successfully. There were only 2 participants that did not - one did not manage to successfully complete 2 of the tasks and other 1 of the tasks - in both cases the reason for failure

was exceeding the expected maximum time. What is worth mentioning is that all participants managed to eventually do the tasks correctly, without the intervention of the moderator.

System Usability Scale results

Table 4.2: SUS questionnaire heatmap

#	Question	A	B	C	D	E	F	G	H	I	J
1	I think that I would like to use this system frequently.	4	3	4	4	4	4	4	4	3	4
2	I found the system unnecessarily complex.	1	1	1	2	2	1	1	1	1	1
3	I thought the system was easy to use.	5	4	5	4	4	5	5	4	4	5
4	I think that I would need the support of a technical person to be able to use this system.	1	1	1	2	2	1	1	1	1	2
5	I found the various functions in this system were well integrated.	5	5	5	3	4	5	4	5	4	4
6	I thought there was too much inconsistency in this system.	1	1	2	1	1	1	1	2	2	2
7	I would imagine that most people would learn to use this system very quickly.	3	3	4	4	4	5	4	5	4	3
8	I found the system very cumbersome to use.	1	4	2	1	1	1	1	1	3	1
9	I felt very confident using the system.	5	4	3	4	2	4	5	4	4	4
10	I needed to learn a lot of things before I could get going with this system.	1	1	1	2	2	1	1	1	2	3

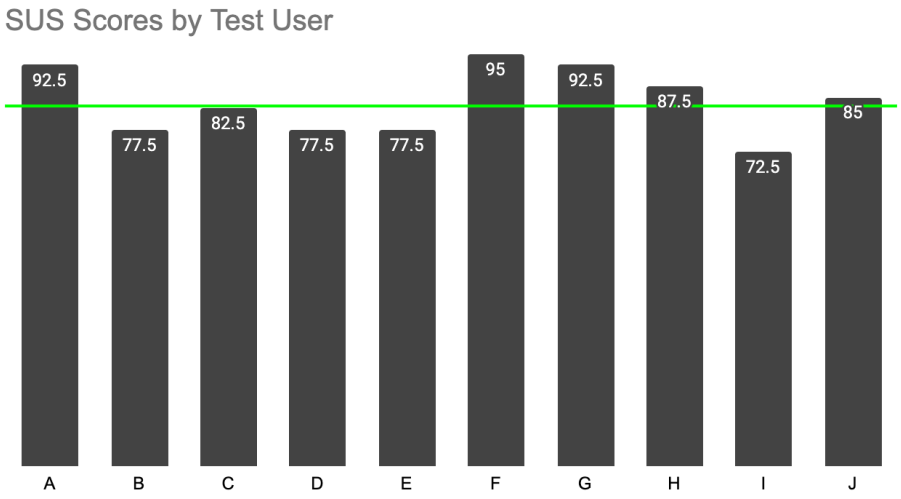


Figure 4.3: SUS scores

The green line represents the average SUS score and its to 87.5

Following Miller's study called "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale" [37], the average score of 87.5

would suggest that usability in our application is "around *Excellent*". However, some of the test subjects were people we knew so we have to assume that their responses might have been skewed in our favor.

4.1.2 Qualitative data

Key observations

After examining our notes and recordings as well as test subjects' comments and responses to open questions we have made the following observations regarding our hi-fi prototype:

1. The drag and drop gesture that needs to first hold an element for a while, even though shown and animated in the help screen, still is not obvious for some of the participants. Instead, they tried tapping and expected that something would happen. We had thought it might still be problematic and the tests confirmed our educated guess.
2. The help screens were definitely a right addition - we did not have to explain "what" and "how" can be done to our test subjects as these screens explained the necessary metaphors. However, their design was not as good as it could have been. Across multiple participants we noticed that it was not clear to them for a brief moment that help screens are just static pages and they tried to interact with it, e.g. by initiating drag and drop gestures there. Nevertheless, all of them noticed their mistake on their own. Participants often pointed out that the dots on the bottom of the screen representing what tab the user is on are hard to spot in the beginning and perhaps should be more prominent.
3. The naming "Alerts" and "Triggers" is confusing, it makes users a bit unsure which button on the main screen they should click to configure the desired thing.
4. Previous observations confirm some of the users comments that in the beginning they had initial difficulties but quickly adapted.
5. We have not observed any significant relation between the age of the participant and corresponding test time or SUS score.
6. Whether the user had had previous experience using Apple devices or not did not influence the course of the test or the final score in a significant way - the same applies to whether the test subject wore glasses or not.

4.1.3 Additional feedback

We also received some feedback that was not observed across many participants but we still regarded it as interesting and worth exploring.

1. One of the users praised the idea that we show a kind of a halo around bubbles in the alerts screen representing configured blink color and mentioned that apart from that they would like to have a tiny icon representing that sound signal is switched on for a given alert.
2. That same user suggested that when you tap on the already selected action variant for a trigger, a popup should be closed.
3. What was also suggested was that we should reconsider usage of drag and drop gestures as there might be issues with it in terms of accessibility.
4. One of the users suggested changing the name of **Settings** screen to **About** as it wasn't clear for them where to look for support contact - they expected to find it in the main menu in the upper right corner. This was probably due to the fact that the other screens have visible **Help** button there.
5. Two users suggested that the buttons visible in the main menu screen should more resemble a button, e.g. by adding some border.

4.1.4 Answering usability study research questions

Having done the analysis, where SUS scores, observations of the moderators and answers given by the participants unanimously showed similar results, with a great deal of confidence we can answer research questions we defined for the purpose of this study.

Is the mobile application usable?

Based on the results, we conclude that the mobile application is usable. Test participants agreed that it has a clean and intuitive interface that makes navigation and interactions very smooth. The design elements and controls are well-organized, allowing users to easily understand and access various features. Additionally, the application provides clear instructions and guides users through different tasks, ensuring a positive user experience.

Is the mobile application self-explanatory and successfully guides the user?

Judging by the results of our study, the mobile application is self-explanatory. Users can easily understand its functionality without needing external as-

sistance. The application incorporates clear and concise labels, icons, and visual cues, making it intuitive for users to navigate through different sections. Furthermore, the application provides contextual hints and prompts to guide users along the way, ensuring they can successfully accomplish their tasks.

Is it easy for the users to configure the necessary settings?

Test results suggest that configuring the necessary settings in the mobile application is easy for the users. Both alerts and triggers screens are well-structured and logically organized, making it effortless for users to locate the specific options they need to customize. The application offers clear explanations and help screens, ensuring users understand the implications of their choices.

4.2 Discussion

In summary, the test confirms that the companion application's development is progressing in the right direction even though there are areas where it could be improved. What is additionally uplifting, is that the test subjects, as well as many of our colleagues, liked the idea of Xocchiali glasses as a whole. Surprisingly, that also includes people that do not wear glasses on a daily basis - they would often say that a pair of sunglasses might be something they would consider.

Nevertheless, the application is only a part of the whole product and in order to evaluate it properly more research should be carried out. For instance, it would be very beneficial to conduct test sessions with both the application and the actual glasses prototype including the onboarding of the user which we did not cover in our study.

During the test sessions we did not use the development board with the microcontroller we were working on as we believed the application for it was not developed enough. Additionally, to properly test the usability of the glasses-application pair as a whole would require much more than short test sessions - there are many corner cases when it comes to the Bluetooth communication, e.g. reconnecting after loss of connection due to a physical distance between the devices. We believe that it would be very hard to create an artificial testing environment to check for different use cases. Perhaps, once the development of the product reaches the point where we can have multiple copies of a working prototype of the glasses frames, it would be a good idea to have a small group of beta testers who would use

the prototypes daily for an extended period of time - this would allow to better understand different scenarios and find more edge cases.

Methodological considerations

Unfortunately, due to time limitations some of the tests were done with one moderator only. Nevertheless, we believe it did not affect the results in any way, as the tests did not require much props and a second computer could be replaced with analog equivalent - pen and paper. We did not feel during such tests that taking the responsibilities of both roles is too much to handle for a single person.

4.2.1 Suggested improvements

Even though the study has shown that the usability of the mobile app is on a satisfying level, it has also shown us that there are some things that can be improved.

Firstly, we suggest changing the button labels in the main menu like so: "Alerts" to "Notifications" - we believe that people associate word "notification" more strongly with being informed by a mobile phone rather than "alert". Moreover, we suggest changing "Triggers" to "Actions" as the word "trigger" does not tell whether it is something that can be triggered (by the user) or is triggered (by some event).

Secondly, another thing to improve is drag and drop interaction. The issue with current implementation is that there is no immediate feedback for the user and therefore there is nothing indicating that one should hold a finger steady for a bit more to interact with an object. We propose to add animated fill of the border when pressing on a bubble, which would start from no border and reach full border fill in the moment when a bubble pops out and is in a state that one is holding it and can manipulate it.

Help screens are what should also be improved. First issue with them is that static elements on help screen pages look like interactive components and people try to use them. We propose to make these elements a bit smaller and add a border around it, which would create an impression more resembling the one of, for instance, a static image. Another issue is that it is not obvious that you can swipe between different pages in the help screen. Right now there are just small dots indicating different pages. We believe that making these indicators more visible and adding additional arrow buttons for navigation between pages would solve the issue. Additionally, it would help a bit with the first issue as well.

These are the improvements we suggest based on the test results. There are some other changes we speculate would be beneficial but we would not like to jump to conclusions and more testing needs to be done to confirm it first.

4.2.2 Possible future work

Even though a lot has been done in terms of the design and implementation of the mobile application, as well the thorough usability tests were conducted, there is still a lot of room for improvement. The first thing that could be done is to implement enhancements suggested in Section 4.2.1 as they are backed by results of the usability study. Moreover, more extensive testing to uncover other possible issues could be done on the mobile application prototype. There is also a part about implementing the rest of the modules for the microcontroller application and integrating it with the mobile application. Such integration would encompass both connecting existing components with remote, Bluetooth data sources and creating new components for pairing with the glasses and overall onboarding activities. Having both parts fully integrated, one should once more conduct a more large-scale user test as there would be a whole new set of interaction challenges that have not been tested yet. Lastly, a port of the application for the Android devices could be developed with all the necessary adjustments on the microcontroller part.

Chapter 5

Conclusions

In this master thesis project we designed and created both low fidelity and high fidelity prototypes of a companion application for smart AR glasses by Xocchiali. After that, we conducted a usability study in order to verify our ideas and implementation. Based on the results, we can conclude that what we have developed is satisfying and could be used as foundations for the actual product in the future. Having done the prototype as an iOS application, the process of transforming it into the final product would be even simpler. Nevertheless, there is still a lot of room for research and improvements in different areas.

Another goal that we met was to program a Qualcomm microcontroller with a proof of concept application, verifying the feasibility of using this hardware in the Xocchiali glasses. In the process we have gathered necessary know-how for the early stage of the development of the product and we developed foundations for the actual product.

Bibliography

[1] Nuri Basoglu, Ali Emre Ok, and Tugrul U. Daim. "What will it take to adopt smart glasses: A consumer choice based review?" In: *Technology in Society* 50 (Aug. 2017), pp. 50–56. DOI: 10.1016/j.techsoc.2017.04.005. URL: https://doi.org/10.1016/j.techsoc.2017.04.005.

[2] Eboni Smith et al. "A comfort analysis of using smart glasses during "picking" and "putting" tasks". In: *International Journal of Industrial Ergonomics* 83 (May 2021), p. 103133. DOI: 10.1016/j.ergon.2021.103133. URL: https://doi.org/10.1016/j.ergon.2021.103133.

[3] Grand View Research. *Wearable Technology Market Share & Trends Report*. Accessed: 04/2023. URL: https://www.grandviewresearch.com/industry-analysis/wearable-technology-market.

[4] Paul Milgram and Fumio Kishino. "A Taxonomy of Mixed Reality Visual Displays". In: *IEICE Transactions on Information and Systems* 77 (1994), pp. 1321–1329.

[5] Borko Furht (eds.) Julie Carmigniani Borko Furht (auth.) *Handbook of Augmented Reality*. 1st ed. Springer, 2011. ISBN: 1461400643; 1461400635; 9781461400639; 9781461400646.

[6] Ronald T. Azuma. "A Survey of Augmented Reality". In: *Presence: Teleoperators and Virtual Environments* 6.4 (Aug. 1997), pp. 355–385. DOI: 10.1162/pres.1997.6.4.355. URL: https://doi.org/10.1162/pres.1997.6.4.355.

[7] Donald A. Norman and Stephen W. Draper. *User Centered System Design; New Perspectives on Human-Computer Interaction*. USA: L. Erlbaum Associates Inc., 1986. ISBN: 0898597811.

[8] Jeffrey. Rubin and Dana. Chisnell. *Handbook of usability testing: how to plan, design, and conduct effective tests*. Indianapolis, Ind.: Wiley Publ., 2008. ISBN: 9780470185483 0470185481. URL: http://www.worldcat.org/search?qt=worldcat_org_all&q=0470185481.

[9] Don Norman and Jakob Nielsen. *The Definition of User Experience (UX)*. Accessed: 06/2023. URL: https://www.nngroup.com/articles/definition-user-experience/.

[10] Ergonomics of human-system interaction Technical Committee ISO/TC 159. Subcommittee SC 4. *ISO 9241-11:2018 Ergonomics of Human-system Interaction: Egonomie de L'interaction Homme-système. Usability : definitions and concepts. Utilisabilité : définitions et concepts. Part 11. Partie 11*. International standard. ISO, 2018. URL: https://books.google.se/books?id=Uc8dygEACAAJ.

[11] John Brooke. "SUS – a quick and dirty usability scale". In: Jan. 1996, pp. 189–194.

[12] Apple Inc. *UIKit*. Accessed: 04/2023. URL: https://developer.apple.com/documentation/uikit.

[13] Apple Inc. *SwiftUI*. Accessed: 04/2023. URL: https://developer.apple.com/xcode/swiftui/.

[14] Steve Heath. *Embedded Systems Design*. 2nd. USA: Butterworth-Heinemann, 2002. ISBN: 0750655461.

[15] Michael Barr and Anthony J Massa. *Programming Embedded Systems*. en. 2nd ed. Sebastopol, CA: O'Reilly Media, Oct. 2006.

[16] Bluetooth SIG, Inc. *Bluetooth protocol specifications*. 2022. URL: https://www.bluetooth.com/specifications/specs/ (visited on 12/2022).

[17] Bluetooth SIG, Inc. *Bluetooth Core Specification*. 2022. URL: https://www.bluetooth.com/specifications/specs/core-specification-5-0/ (visited on 12/2022).

[18] Bluetooth SIG, Inc. *GATT Suplement Specification*. 2022. URL: https://www.bluetooth.com/specifications/specs/gatt-specification-supplement/ (visited on 12/2022).

[19] Apple. *ANCS specificaiton*. Accessed: 03/2023. URL: https://developer.apple.com/library/archive/documentation/CoreBluetooth/Reference/AppleNotificationCenterServiceSpecification/Specification/Specification.html.

[20] Kent Beck et al. *Manifesto for Agile Software Development*. 2001. URL: http://www.agilemanifesto.org/ (visited on 12/2022).

[21] Kai Petersen, Claes Wohlin, and Dejan Baca. "The Waterfall Model in Large-Scale Development". In: *Product-Focused Software Process Improvement*. Ed. by Frank Bomarius et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 386–400. ISBN: 978-3-642-02152-7.

[22] Apple Inc. *Human Interface Guidelines*. Accessed: 04/2023. URL: https://developer.apple.com/design/human-interface-guidelines/.

[23] Apple Inc. *Xcode*. Accessed: 05/2023. URL: https://developer.apple.com/xcode/.

[24] Scott Chacon and Ben Straub. *Pro git: Everything you need to know about Git*. English. Second. Apress, 2014. URL: https://git-scm.com/book/en/v2.

[25] Inc. Github. *GitHub*. 2020. URL: https://github.com/.

[26] Marc J. Rochkind. "The source code control system". In: *IEEE Transactions on Software Engineering* SE-1.4 (1975), pp. 364–370. DOI: 10.1109/TSE.1975.6312866.

[27] Atlassian. *Trello - a project management tool*. Accessed: 05/2023. URL: https://trello.com/.

[28] John M Gross and Kenneth R McInnis. *Kanban made simple: Demystifying and applying Toyota's legendary manufacturing process*. en. Amacom Books, 2003.

[29] Marian H.H. Willeke. "Agile in Academics: Applying Agile to Instructional Design". In: *2011 Agile Conference*. 2011, pp. 246–251. DOI: 10.1109/AGILE.2011.17.

[30] Apple. *iPhone 14 - Technical Specifications*. Accessed: 05/2023. URL: https://support.apple.com/kb/SP873?locale=en_US.

[31] Apple. *iPhone SE (2nd generation) - Technical Specifications*. Accessed: 05/2023. URL: https://support.apple.com/kb/SP820?locale=en_US.

[32] John Kounios and Mark Beeman. "The iAha!/i Moment". In: *Current Directions in Psychological Science* 18.4 (Aug. 2009), pp. 210–216. DOI: 10.1111/j.1467-8721.2009.01638.x. URL: https://doi.org/10.1111/j.1467-8721.2009.01638.x.

[33] Qualcomm. *Qualcomm QCC3071*. Accessed: 04/2023. URL: https://www.qualcomm.com/products/internet-of-things/consumer/audio/qcc30xx-series/qcc3071.

[34] Qualcomm. *Qualcomm QCC5171*. Accessed: 04/2023. URL: https://www.qualcomm.com/products/internet-of-things/consumer/audio/qcc5100-series/qcc5171.

[35] Microsoft. *Swift Pair*. Accessed: 05/2023. URL: https://learn.microsoft.com/en-us/windows-hardware/design/component-guidelines/bluetooth-swift-pair.

[36] Anna Borkowska and Katarzyna Jach. "Pre-testing of Polish Translation of System Usability Scale (SUS)". In: vol. 521. Sept. 2017, pp. 143–153. ISBN: 978-3-319-46582-1. DOI: 10.1007/978-3-319-46583-8_12.

[37] Aaron Bangor; Philip Kortum; James Miller. "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale". In: *Journal of Usability Studies* 4.3 (2009), pp. 114–123.

Appendix A

Task list

No.	Task	Subtasks	Success condition	Est. time [min]
1	Configuration of alerts	1. Configure call alert 2. Configure call alert from all others 3. Configure app alert 4. Change alerts placement 5. Remove alert 6. Edit alert	All subtasks completed	6
2	Configuration of triggers	1. Configure media control actions 2. Configure call action 3. Delete actions 4. Change placement of an action 5. Add non-configurable action	All subtasks completed	5
3	Miscellaneous activities	1. Contact support 2. Find paired glasses information 3. Disconnect glasses	All subtasks completed	2

Appendix B

Briefing script

B.1 English version

Xocchiali wants to develop a product - smart glasses. The idea is to have a limited set of capabilities but in turn make them look like any other frames that you can choose at your optician's shop, not cause fatigue when wearing and have long lasting battery. In short, they could be your main pair of glasses that you use daily, throughout the whole day, instead of just being a gadget. They would have speakers, or rather a bone conducting system, and a microphone. That would make them basically just another Bluetooth headphones but integrated into glasses. But what's actually special about them is that they will also have an LED light for notifications you configure and a button which triggers configurable actions! All of the configuration takes place in the mobile app. And that app is exactly the object of this test.

As for the test will be conducted, what you do on the phone, as well as audio of our conversation will be recorded. In the beginning I'll ask you a few questions. Then you will be given a scenario with description of tasks you're supposed to do. When doing the tasks, if you think completed what you were supposed to do, just continue and don't ask me for confirmation. If you think that the instructions are unclear please do ask me for help. Overall, you can ask me any question during the test because of e.g. technical issue with the device, but I might not answer all of them as you're supposed to do the tasks on your own without my help.

In the end, after you're done with it, I'll ask you some questions again. Everything should take around 15 minutes.

Here's the consent form that once again states that you're going to be recorded and that we keep some data about you, that is your age, gender

and experience in relevant area. Mind that at any point of the test you can simply stop it and after the test you can always withdraw your participation. Such action means that we will destroy all the data about you gathered during the test.

Also if you want to have a snack or drink some juice feel free to take it at anytime during the test or after it.

B.2 Polish version

Xocchiali chce opracować produkt - smart okulary. Pomysł polega na tym, aby miały one ograniczony zestaw funkcjonalności, ale z kolei wyglądały jak każde inne zwykłe oprawki, które możesz normalnie wybrać u swojego optyka, nie powodowały zmęczenia podczas noszenia i długo działały bez konieczności ładowania. W skrócie, mogłyby być twoją główną parą okularów, których używasz codziennie przez cały dzień, zamiast być tylko gadżetem. Miałyby głośniki, lub właściwie system przewodnictwa kostnego, oraz mikrofon. To sprawiłoby, że byłyby to właściwie po prostu słuchawki Bluetooth, ale zintegrowane z okularami. Ale to, co jest naprawdę wyjątkowe, to fakt, że będą również miały diodę LED do powiadomień, które można skonfigurować oraz przycisk, który uruchamia konfigurowalne akcje! Cała konfiguracja odbywaja się przez aplikację mobilną. A ta aplikacja jest dokładnie tym co jest obiektem tego testu.

Co do sposobu przeprowadzenia testu, to to, co robisz na telefonie, oraz nasza rozmowa, będą nagrywane. Na początku zadam ci kilka pytań. Następnie zostanie ci przedstawiony scenariusz z opisem zadań, które masz wykonać. Wykonywując zadania, jeśli uważasz, że zrobiłeś to, co miałeś zrobić, po prostu kontynuuj i nie proś mnie o potwierdzenie. Jeśli uważasz, że instrukcje są niejasne, poproś mnie o pomoc. Ogólnie rzecz biorąc, możesz zadawać mi dowolne pytanie podczas testu, na przykład o w razie gdyby nadarzyły się jakieś problemy techniczne z telefonem, ale niekoniecznie na wszystkie odpowiem, ponieważ powinieneś wykonać zadania samodzielnie, bez mojej pomocy.

Na końcu, gdy skończysz, zadam ci kilka kolejnych pytań. Całość powinna zająć około 15 minut.

Oto formularz zgody, który ponownie mówi o tym, że będziesz nagrywany i że będziemy przechowywać pewne dane dotyczące twojego wieku, płci i doświadczenia w paru kwestiach. Pamiętaj, że w dowolnym momencie testu możesz po prostu go przerwać, a po zakończeniu testu zawsze możesz wycofać swoje uczestnictwo. Takie działanie oznacza, że zniszczymy wszystkie

dane o tobie zgromadzone podczas testu.

Jeśli chcesz coś przekąsić lub wypić soczku, możesz zrobić to w dowolnym momencie trwania testu lub również po jego zakończeniu.

Appendix C

Instructions

C.1 English version

Background: You just got newest Xocchiali glasses, opened the companion app on your smartphone and completed the pairing process.

1. You'd like to configure your application in such a way that:
   (a) If contact "Mama" calls you, your glasses should blink twice with a red color without a sound.
   (b) If any other contact calls you, it should blink once with blue color but shouldn't produce any sound.
   (c) If you receive a Messenger notification, it should produce a piano sound without any blinking.
   (d) You didn't like how you placed the alerts and you'd like them all to occupy last row only.
   (e) You've decided that you don't want to be notified if "Mama" calls you, remove this alert.
   (f) Also, you'd like to edit Messenger notification so that the glasses additionally blink once with blue light.
2. You remember that the glasses also advertised that you can configure your own actions TRIGGERED by the push of the button. Try to find in the application where you can configure that.
   (a) You want to make it so you can play and pause your music with a single push a button. Configure it in such a way that it would work like that.

(b) Now that you have play/pause function you think that it's only natural to make it so you can skip the song on double click. Make it work like that.

(c) You often find yourself calling your dad. Make it so that triple clicking the button would result in making a call to the contact named "Tata".

(d) After giving it more thought, you realise that you wouldn't use your glasses for music anyway and therefore you don't want your precious trigger slots being occupied by actions related to it. Since slots for single and double click are free to use you want to move the action that calls your dad to double click slot and add new action on single click that would make your phone ring so that you can easily find it. Summing up: single click - find phone, double click - call dad, triple click - nothing.

3. You've got several questions about your glasses model and you'd like to contact support.

4. You've contacted the support and they'd like you to tell them what's the serial number of you glasses, so you'd like to find it.

5. They ask you to disconnect the glasses and connect them once again.

C.2 Polish version

Informacje wstępne: Właśnie otrzymałeś najnowsze okulary Xocchiali, otworzyłeś aplikację do nich na swoim smartfonie i ukończyłeś proces parowania.

1. Skonfiguruj swoje okulary poprzez aplikację w taki sposób, żeby:
   (a) Jeśli kontakt "Mama" do Ciebie zadzwoni, Twoje okulary powinny zamigać dwa razy czerwonym światłkiem bez żadnego dźwięku.
   (b) Jeśli jakikolwiek inny kontakt do Ciebie zadzwoni, okulary powinny zamigać raz niebieskim światłkiem, również bez dźwięku.
   (c) Jeśli otrzymasz powiadomienie z aplikacji Messenger, powinno to wywołać dźwięk fortepianu, bez migania.
   (d) Nie podoba Ci się, jak umieściłaś alerty i chcesz, żeby wszystkie zajmowały tylko ostatni rząd (pierwszy jeśli już wszystkie są w ostatnim).
2. Zdecydowałeś(-łaś), że nie chcesz być powiadamiany(-ana), jeśli "Mama" do Ciebie zadzwoni, usuń to powiadomienie.
3. Chciał(a)byś również edytować powiadomienie z Messengera tak, aby okulary dodatkowo zamigły raz niebieskim światłem.
4. Pamiętasz, że okulary reklamowały możliwość skonfigurowania własnych akcji, które będą wywoływane poprzez naciśnięcie przycisku.
   (a) Chcesz, aby przycisk umożliwiał Ci odtwarzanie i zatrzymywanie muzyki po pojedynczym naciśnięciu. Skonfiguruj okulary, żeby działały w ten sposób.
   (b) Teraz, gdy masz funkcję odtwarzania/pauzowania, uważasz, że naturalnym jest, aby móc pominąć piosenkę podwójnym kliknięciem. Skonfiguruj okulary, żeby działały w ten sposób.
   (c) Często dzwonisz do swojego taty. Skonfiguruj przycisk w taki sposób, żeby po potrójnym kliknięciu dzwonić do kontaktu o nazwie "Tata".
   (d) Po namyśle dochodzisz do wniosku, że i tak nie będziesz używał swoich okularów do słuchania muzyki, więc nie chcesz, żeby Twoje cenne pojedyncze i podwójne klikniącia przycisku były zajęte przez związane z tym akcje. Ponieważ miejsca dla pojedynczego i podwójnego kliknięcia są wolne do użytku, chcesz

przenieść akcję dzwonienia do taty na miejsce podwójnego kliknięcia i dodać nową akcję na pojedyncze kliknięcie, która spowoduje dzwonienie i wibrowanie telefonu, żeby ułatwić go znajdowanie. Podsumowując: jedno kliknięcie - szukaj telefonu, podwójne kliknięcie - dzwoń do taty, potrójne kliknięcie - nic.

5. Masz kilka pytań dotyczących swojego modelu okularów. Skontaktuj się z pomocą techniczną.
6. Skontaktowałeś się z pomocą techniczną i chcą, abyś podał numer seryjny swoich okularów, więc chcesz go znaleźć.
7. Poproszono Cię, aby odłączyć okulary i podłączyć je ponownie.

Appendix D

Introductory questions

D.1 English version

1. What is your age?
   - 20 - 24
   - 25 - 29
   - 30 - 34
   - $>$ 34
2. What is your gender?
3. Where are you from?
4. Are you wearing glasses?
5. How easy is it for you to use smartphones, computers, technologies that are new for you?
   - Very easy
   - Rather easy
   - Neither easy nor hard
   - Rather hard
   - Very hard
6. Do you use, or have used in the past, any Apple devices? If so, which ones and approximately for how long?
7. Do you know what Apple's Shortcuts are? If so, do you have experience using them.

8. Do you have any experience using smart wearables like watches, bracelets, glasses etc? If so, which ones and approximately for how long?

D.2 Polish version

1. Ile masz lat?
   - 20 - 24
   - 25 - 29
   - 30 - 34
   - $>$ 34
2. Jaka jest twoja płeć?
3. Skąd pochodzisz?
4. Czy nosisz okulary?
5. Jak oceniasz to jak łatwe jest dla ciebie korzystanie ze smartphonów, komputerów, nowych dla ciebie technologii??
   - Bardzo łatwe
   - Raczej łatwe
   - Ani łatwe ani trudne
   - Reczej trudne
   - Bardzo trudne
6. Czy używasz lub używałeś/aś w przeszłości jakiś urządzeń Apple? Jeśli tak to jakich i jak długo?
7. Czy wiesz czym są Skróty Apple? Jeśli tak to czy masz jakieś doświadczenie w korzystaniu z nich?
8. Czy masz jakieś doświadczanie w korzystaniu z smart wearables (urządzenia ubieralne), takich jak smartwatche, bransoletki, okulary itp? Jeśli tak to jakie i jak długo?

Appendix E

Concluding questions

E.1 English version

- Standard usability scale:
  1. I think that I would like to use this system frequently.
  2. I found the system unnecessarily complex.
  3. I thought the system was easy to use.
  4. I think that I would need the support of a technical person to be able to use this system.
  5. I found the various functions in this system were well integrated.
  6. I thought there was too much inconsistency in this system.
  7. I would imagine that most people would learn to use this system very quickly.
  8. I found the system very cumbersome to use.
  9. I felt very confident using the system.
  10. I needed to learn a lot of things before I could get going with this system.

  Likert scale used:

  1. Strongly Disagree
  2. Disagree
  3. Neutral
  4. Agree

5. Stongly Agree

- Our open questions:
  - What are your overall impressions?
  - Did you find the interface intuitive in general?
  - Was there anything about the interface that felt weird/out of place etc.?
  - What task or tasks did you find most challenging (if any)?
  - [Any other things that we want to ask depending on how the test went]
  - Do you have any other comments?

E.2 Polish version

- Standard usability scale:
  1. Myślę, że często używałbym/używałabym tego systemu.
  2. Odbieram ten system jako niepotrzebnie skomplikowany.
  3. Myślę, że system jest łatwy w użyciu.
  4. Myślę, że potrzebowałbym/potrzebowałabym wsparcia asystenta, aby używać tego systemu.
  5. Uważam, że różne funkcje tego system są dobrze zintegrowane.
  6. Myślę, że w tym systemie jest za dużo niespójności.
  7. Wydaje mi się, że większość ludzi nauczyłaby się bardzo szybko używać tego systemu.
  8. Odbieram ten system jako bardzo niewygodny w użyciu.
  9. Czułem/czułam się bardzo pewnie używając tego systemu.
  10. Musiałem/musiałam nauczyć się wielu rzeczy, zanim zacząłem/zaczęłam właściwie posługiwać się tym systemem.

  Likert scale used:

  1. Zdecydowanie nie zgadzam się
  2. Raczej się nie zgadzam
  3. Nie mam zdania
  4. Raczej się zgadzam
  5. Zdecydowanie się zgadzam

- Our open questions:
  - Jakie są twoje ogólne wrażenia?
  - Czy według ciebie interfejs był intuicyjny?
  - Czy coś w aplikacji sprawiało dla ciebie wrażenie dziwnego, źle usytuowanego itp?
  - Które zadania były dla Ciebie najbardziej wymagające (chyba że żadne)?
  - [Any other things that we want to ask depending on how the test went]

– Czy chcesz coś jeszcze dodać?