Exploring User Experience and Control in IoT Environments through Assisted Reality

Carl Rutholm

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

MASTER THESIS

Exploring User Experience and Control in IoT Environments through Assisted Reality

Carl Rutholm

June 10, 2023

Master's thesis work carried out at Google Cloud.

Supervisors: Günter Alce, Jakob Ehrl and Peter Hedberg

Examiner: Joakim Eriksson

Exploring User Experience and Control
in IoT Environments using Assisted Reality

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

Subject: Interaction Design MAMM01
Division: Ergonomics and Aerosol Technology
Supervisors: Günter Alce, Jakob Ehrl and Peter Hedberg
Examiner: Joakim Eriksson

Abstract

Internet of things is a rapidly expanding area with new exciting devices being developed each year. Traditional Internet of Things devices are usually controlled through a smart phone which can be exhausting and overwhelming with multiple applications needed to monitor a set of devices. This thesis explores an alternative technology platform and a single accompanying application to discover, interact and monitor smart devices. An application was developed for Google Glass to discover and control a Philips Hue light bulb and a Sonos WiFi speaker. Glass and the developed application were tested and evaluated against an iPad by twelve test subjects. Semi-structured interviews were held and two different quantitative tools were used to determine which solution subjects favored. The results show that Google Glass is perhaps not the right tool to use today, but show potential to become a preferred technology in the future.

Keywords: Assisted Reality, Internet of Things, Smart device, Google Glass

Sammanfattning

Internet of Things är ett teknikområde som växer i rask takt och utvecklar nya uppfinningar varje år. Traditionell Internet of Things styrning sker oftas via en smart phone och kan vara en besvärlig och överväldigande process där flera olika applikationer kan behövas för att ha tillgång till sina elektroniska enheter. Detta projekt har utforskat en alternativ teknologi för att upptäcka och kontrollera smarta enheter. En applikation utvecklades för Google Glass för att upptäcka och kontrollera en Philips Hue lampa och en Sonos Play:1 trådlös högtalare. Glass och den framtagna applikationen testades och utvärderades gentemot en Apple iPad av tolv försökspersoner. Semi-strukturerade intervjuer genomfördes och två olika kvantitativa verktyg användes för att fastställa vilken lösning testpersonerna föredrog. Resultaten visar att Google Glass kanske inte är det rätta verktyget att använda idag, men har potential att bli framtidens föredragna teknologi.

Acknowledgements

First and foremost I would like to thank my wonderful supervisor Günter Alce at the Department of Design Sciences, LTH. For all the guidance and support he provided me throughout the project with inspiration, motivation, and most importantly all his radiant positivity and kindness.

I would like to thank Jakob Ehrl and Peter Hedberg at Google Cloud for making this collaboration possible. I consider myself very lucky to have been able to write my thesis at Google and I owe this to Jakob and Peter. It has been very pleasant working with you and the company.

I would also like to acknowledge everyone who participated in my user tests that provided me with valuable data I was able to use in my research. Thank you so much for your time and participation.

Lastly I would like to express my deepest gratitude to my family and friends for their love and support throughout my years in Lund and LTH. It has been a long journey and I am happy to have been able to share with you. Thank you so much.

Contents

Chapter 1
Introduction

This chapter briefly introduces the background for the thesis. A short description of the company this thesis was written in collaboration with is presented, followed by the purpose and goals along with scope and limitations, and previous research.

1.1 Background

Smart electronics and devices are getting more popular and can be found in every day life. Most modern devices can be connected to a network and communicate with other devices on the same system, and some can even reach further to other networks using the internet. With wireless home networks and the Internet of things (IoT) movement, wireless control is easier for consumers, and recent estimations predict that the number of smart devices will grow significantly over the upcoming years [2].

Smart assistance's such as Google Assistant and Amazon's Alexa is a popular wireless controlling device that is based on voice commands and home networks. Since it has made its way into households to aid users in their private life, smart home devices have started to become more popular and common to aid users every day scenarios. Smart home devices can be controlled using this technology, but can also be monitored using a smart phone application. However, these applications can be as many as the number of companies developing the products, and become very overwhelming. With the recent advances made in smart glasses technology, focus could shift from smart phones to smart glasses, making them the definitive choice for smart device control. This could be the next step in the tech evolution that will give rise to solutions for both industries, corporate work and private life.

Smart glasses are already being used in many areas of work. For example, people working industrial jobs can use a pair to aid them with guidance through a complicated process [27]. Another example is within health care and surgery, where doctors can monitor vital signs and critical information on their glasses, without taking their eyes off the patient [54]. Since smart glasses offer easy accessible information by placing basically anything computer generated in the users field of view, they have potential to be used for various purposes. One of these purposes could be wireless control.

Imagine being able to step into a control room full with machinery and devices and with one glance across the floor, change settings and decide the entire work process to whatever desired. Imagine having both hands caught up in work and having the need to interact with something far away and with just a look, accomplish this without having to free up any hands and walk over to interact. These future scenarios is something this project will try to contribute to and work towards.

1.2 Google Cloud

Google LLC, commonly referred to as Google, is a multinational technology company based in California that specializes in internet-related services and products. The company was founded in 1998, offering their search engine service for searching the internet. Ever since, Google has expanded and branched out to offer multiple products such as cloud solutions, mobile software, internet services and smart glasses [19].

Google released their smart glasses "Glass" in 2013 with a focus on hands-free information access and have since released a few models, with their most recent in 2019 called Glass Enterprise 2 edition.

This thesis has been made in collaboration with the Google Cloud department in Stockholm which offers cloud based solutions. Among these are AI and Machine Learning services, Compute Engines, and IoT solutions to name a few.

1.3 Purpose and goal

The purpose of this thesis is to explore an alternative way for users to wirelessly interact with smart devices. Google Glass was chosen as the device to use and to evaluate what potential smart glasses could have functioning as a wireless controller for smart devices. An application to interact with smart devices will be developed to let users discover and interact with available devices in the area. This application, as well as the chosen pair of smart glasses, will then be evaluated from a usability and work load point of view, with human user tests to discover their strengths and weaknesses.

The main goal for this thesis is to:

- Evaluate Google Glass as a control platform for smart devices.
- This will be achieved by exploring the following research questions:
  - How well do Google Glass function as a smart device controller from a hardware standpoint?

– How intuitive is a smart glass control application and interface compared to a tablet application?
– How can a user discover what devices in an environment are available and interactable?
– What do people think of smart glasses and their future potential?

1.4 Scope and limitations

Implementing a communication application with desired characteristics and functionalities takes time. The development process can be quite time consuming depending on what is prioritized and because of the 20 week time period given for this Master Thesis, this project had to be limited.

The scope of this master thesis will not include:

- Development for other control units than the chosen pair of smart glasses to whom the application will be exclusively developed for. Other devices might function well with the application but it is not of this thesis concern.
- A universal application to control most or every smart device on the market. Since smart devices run on different communication protocols, only a few smart devices will be included to meet the requirements of this thesis purpose.

1.5 Global goals

In 2015, the General Assembly proposed the Agenda for Sustainable Development with the purpose of guiding towards a more sustainable and inclusive future for all humans. The agenda is set to be reached in 2030 and consists of 17 goals:

- 1. No poverty
- 2. No hunger
- 3. Good health and well-being
- 4. Quality education
- 5. Gender equality
- 6. Clean water and sanitation
- 7. Affordable and clean energy
- 8. Decent work and economic growth
- 9. Industry, innovation and infrastructure
- 10. Reduced inequality
- 11. Sustainable cities and communities
- 12. Responsible consumption and production
- 13. Climate action
- 14. Life below water
- 15. Life on land
- 16. Peace and justice strong institutions
- 17. Partnerships to achieve the goal

This thesis will contribute to goal number 9. Contributions met by goal number 9 will be by promoting inclusive industrialization, enhance research and industrial technologies, and increase access to information and communications technology.

1.6 Related work

This thesis and its purpose drew inspiration from previous research on similar topics. Four main theses were studied:

- Ubi-compass: IoT interaction with wearables [47]
- Control your home with augmented reality [53]
- Using AR gaze-based interaction to control IoT devices [3]
- Smart Homes: Design & Development of an Application From a User Centered Perspective [6]

Ubi-Compass evaluates the possibility to use a "wearable" (smart watch) as a control unit to interact with smart home devices. An application was developed for the wearable and tested on human test subjects to evaluate functionality and discoverability of available smart devices. The wearable was compared to an existing mobile application and the results show that test users generally preferred the wearable when interacting with smart home devices, especially when completing simpler tasks [47].

Control your home with augmented reality explores the potential to use Augmented Reality (AR) on a smart phone to discover and interact with available smart home devices. A smart phone application was developed and tested on human test subjects and the results concluded that AR has the potential to work to control IoT devices but that the technology might not be ready in its current state [53].

Perhaps the most similar topic compared to this thesis is *Using AR gaze-based interaction to interact with IoT devices*. The authors developed an application running on the AR glasses model Magic Leap that track user eye movements and gaze to use as a way of controlling and interacting with IoT devices. Their conclusion was that the method of interaction was viable and usable but that the hardware at the time was too inconvenient to use as an everyday tool [3].

In *Smart Homes: Design & Development of an Application From a User Centered Perspective*, an already existing communication interface was evaluated and improved by analyzing and applying user opinions and data to their development and re-designing the interface with a user centered goal in mind. Their results gave valuable information to what users might appreciate and value in an interface used to control IoT devices [6].

From the valuable information obtained from past research, this thesis was planned out and executed to answer the research questions and reach its purpose and goal.

Chapter 2
Theory

This chapter describe theoretical terms and technical background for this thesis, such as Internet of things, the chosen hardware and software, as well as communication protocols. This is followed by describing the methods of data collection used. These concepts are what the thesis is built upon and supports the reasoning behind the choices made in this project.

2.1 Internet of things

Internet of Things (IoT) is a technical concept that describes electronic devices that can use networks as an information sharing platform. It is all about connecting devices and sensors to the internet or a local network and making them accessible. The idea is to create a network where people and devices can interact with each other seamlessly. Therefore, any device that can be connected to a network and share its information through it can be considered an IoT device [32].

According to some, the movement originated in the 1980s when a coke machine was monitored through a local network by a group of students. However, it is in recent times where IoT has really taken off. The first smart fridge was announced in 2000, the iPhone in 2007 and the number of connected devices exceeded the number of people on earth in 2008 [32]. As of today (2023), there are about 15.14 billion connected IoT devices (just short of twice the total current number of people on earth) and several studies suggests that this number will grow substantially over the next few years [12].

Since the criteria for an IoT device is so simple, it makes the use cases many. One large consumer market is smart devices for so called "smart homes". A smart home is a convenient home setup where electronic devices are monitored and controlled via a networked device [21]. This is directed towards private consumers and some popular devices for this market are Amazon Alexa, Google Chrome Cast, Sonos and Philips Hue.

2.2 Extended reality

Extended Reality is a term to define immersive technology that works with computer generated content and the real world. Extended Reality (XR) can be described using a spectrum of how much a technology uses the real world compared to virtual computations (see figure 2.1) [55]. At the left end of the spectrum, reality is defined which is what would be observed and interpreted when regarding the real world without any digital or graphical impact. On the other far side of the spectrum, Virtual Reality (VR) is defined as the opposite, where the virtual world is observed and interpreted without any impact from the real world. Everything in between can fit into a mixed concept where both the real and virtual world blend together.

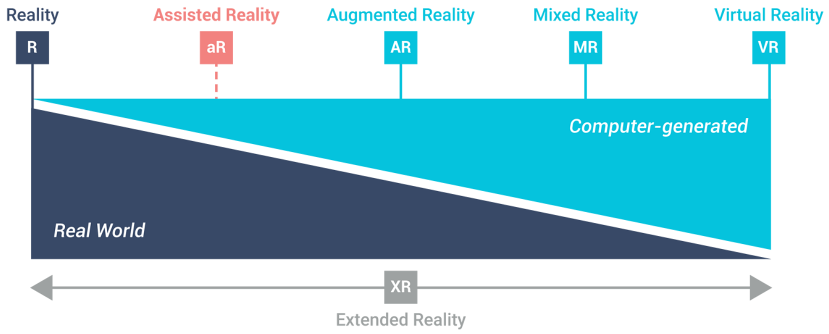


Figure 2.1: The Extended Reality spectrum [55]

Assisted Reality (aR) and Augmented Reality (AR) are two similar technologies that differs from each other in some ways and should not be regarded as the same.

Assisted Reality, as the name suggests, is used primarily to aid and assist the user with information. The technology prioritizes the real world and uses digital information as second priority without trying to blend virtual content into the real world for an interactive experience [55]. For instance, if a construction worker are having troubles using both hands on heavy machinery while needing to follow a written guide or plan, the worker could use aR to get the needed information directly into the field of view. This would free up the worker's hands that otherwise could be occupied with any.

Augmented Reality introduces computer generated content to the real world to give the illusion that they exist in the same space. AR can rely both on the real world or the virtual world as the base and bring in virtual and real elements on top to enhance a users experience [4]. Popular examples are the AR game Pokemon Go and the AR furnishing guide IKEA Place, where users can use their smart phone camera to see both the real world in front of them as well as virtual animals or furniture that is rendered on the screen in real time [37][26].

Aspects from both aR and AR will be explored in this thesis.

2.3 Discoverability

Discoverability, within the context of human-computer interaction (HCI), refers to the extent to which users can easily and intuitively determine the available actions, features, and functionalities of a system or interface. It focuses on enabling users to locate, identify, and understand how to interact with the various elements of a digital system. A highly discoverable interface allows users to quickly grasp the system's capabilities, explore its features, and efficiently perform desired actions without undue effort or confusion [31].

The concept of discoverability was popularized by Don Norman, a prominent figure in the field of HCI. Norman emphasizes the importance of discoverability in creating user-friendly interfaces that support effective interaction. He poses the fundamental question, "Is it possible to even figure out what actions are possible and where and how to perform them?", which can be considered central in user interface design [38].

2.4 Smart Glasses

Smart glasses is a term to describe glasses with embedded "smart" functions. Smart functions and the smartness of smart glasses can be defined as how well glasses use computer hardware, software, mobility, wirelessness and the ability to connect to other devices and services [57]. In order for a pair of glasses to fit this description, they will need to have some kind of computing power and functionality.

2.5 Hardware

2.5.1 Google Glass Enterprise Edition 2

Google Glass is a pair of smart glasses developed by the tech company Google LLC (see figure 2.2). Glass consists of a computer and an optical head-mounted display. The computer view is projected onto a rectangular prism located in the users upper-right view. The user can navigate the computer using a touch pad on the right hand spectacle frame. Glass also has a microphone input, audio output, an integrated camera, WiFi and Bluetooth protocols and various sensors such as 3-axis accelerometer, gyroscope and magnetometer. Glass run the operative system Android Open Source Project 8.1 (Oreo) and are able to run Java applications and software [15]. Glass Enterprise Edition 2 was officially released on March 23, 2022 and no regular updates and support are scheduled after September 15, 2023 [17].

Figure 2.2: Google Glass Enterprise Edition 2 [51]

2.5.2 Sonos Play:1

Play:1 is a wireless smart speaker from Sonos (see figure 2.3). The speaker can be controlled through a wireless network and has integrated music services such as Spotify and internet radio [49]. Sonos also offer their own REST API to integrate speaker controls to applications and programs [48].

Figure 2.3: Sonos Play:1 [50]

2.5.3 Philips Hue

Hue is a smart lighting series developed by Philips (see figure 2.4). The series consists of a Bridge, several different light sources (bulbs, strips and lamps), sensors and software that can be controlled wirelessly using a network. The Bridge is the center of communication and connects to the light sources using a Zigbee protocol. Connecting the bridge to a network using an Ethernet cable gives users on the same network access to interact with the light sources wirelessly by sending commands to the Bridge using the Philips Hue app or by the Philips REST API service [41].

Figure 2.4: Philips Hue bridge and light bulbs [42]

2.5.4 Raspberry Pi

Raspberry Pi is a series of small single board computers made by the British Raspberry Pi foundation (see figure 2.5). The computers consists of a single board with enough hardware to run programs and operative systems such as Linux, as well as several components to use as inputs and outputs. The computers were made with the purpose of learning children and young people about computers and programming. Different models of Raspberry Pi have different components and the one used in this thesis is the Raspberry Pi 3 model B+. The most important components on model B+ are its several USB-ports, an HDMI port, an Ethernet port, a WiFi receiver as well as multiple programmable pin-inputs and outputs [44][39].

Figure 2.5: Raspberry Pi 3 model 3+ [43]

2.6 Software

2.6.1 Home Assistant

Home Assistant is an open source software that works as a home automation program. Home Assistant integrates and connects devices that are set up to a network and bridges the gap in between them by offering a communication hub where the devices can communicate. It also offers an interface for users to control all connected smart devices from one place. Users can program the Home Assistant to set up unique automations and settings to serve certain purposes and to execute specific tasks. In this way devices can be set up to exchange information, respond to each other, and work together in a fully integrated wireless environment. Home Assistant supports, as of when this thesis is written, over 2400 integrations for different sensors and protocols making it a versatile and powerful software. Because of the open source approach, developers are constantly working on new integrations and increasing the number of devices that can be added to the interface [9].

Home Assistant can be set up wirelessly or using an Ethernet cable to a local network. Using the local network it can discover other smart devices that are on the same network. Home Assistant saves the devices information and adds them to its collection of devices, called entities.

An entity is a representation of the saved device that follows a standardized structure with predefined services that can be executed. For example, most light entities can be switched on and off, and this predefined service can be executed using the predefined service **light.turn_off**. However, if a device have special attributes or functions, for example Philips Hue's scenery presets, these can be specialized as a unique service either in the specific Philips Hue device integration, or by a user in their private Home Assistant software. In the context of Hue, this non-standardized service can be activated by **hue.activate_scene**.

According to the Home Assistant developers, the software is best suited to run on a remote Raspberry Pi. Users can access the Home Assistant interface by logging in via a web browser, using the Home Assistant app for iOS and Android or using the Home Assistant REST API. [22]

2.7 Communication

2.7.1 HTTP communication

Hypertext Transfer Protocol (HTTP) is a protocol used in web communication to transfer files from a server to a client. A client requests a file from the server and the server responds with either the requested file or with an error indication. The request is composed of a request method, request URI, message header and potential body [30].

HTTP Requests

There are many request methods but common ones are:

- GET - Retrieve resource
- POST - Create a new resource
- PUT - Overwrite an existing resource or create new if it does not exist
- DELETE - Delete a resource

Application designers can specify what the methods do. POST can behave in a similar way as PUT to make applications rely on GET and POST for most requests, which is common when sending requests to APIs [1].

The request URI (Uniform Resource Identifier) is an identification for resources and services that can be accessed by request. This specifies what the requester is trying to access and usually follows a similar design for similar resources or servers, hence the term "Uniform". The URI syntax was designed as a sequence of characters from the letters of the Latin alphabet, digits and a few special characters. This could be organized to form anything from a web address to access a web page, to a mailto scheme from electronic mail addresses, for example "mailto:example@example.com" [5].

Finally, the message header of the HTTP request defines the properties of the request and specifies the context for the server. This gives more depth to the requests and give additional information such as authentication, what data is being transferred, or what response type should be sent back [5]. This is done by setting the Content-type header which, for example, could be "Content-type: image/jpg" in order to specify that a jpg-image will be requested. The header can optionally be followed by a body to specify additional data that is being sent with the request. This is usually the case when using the method POST since the nature of the request is to create or update a resource. The new data to place in the resource is put in the body and processed by the server receiving the request [8].

HTTP Response

An HTTP response is a message the server is sending back to the requesting client after a request. Its purpose is to return information to the client about how the request was processed. The response usually consists of:

- A status line
- HTTP headers
- Message body

The status line contains the HTTP protocol version number, status code and status text. The protocol version number inform what version is being used and the status code and text indicate how the request was processed. The status code is a three digit number and can be divided into four categories:

- 1xx - Informational: The request was received and the process continues
- 2xx - Success: The request was successfully received, understood and accepted
- 3xx - Redirection: The request was incomplete and further action is needed in order to complete the request
- 4xx - Client error: The request contains bad syntax and/or cannot be fulfilled
- 5xx - Server error: The server failed to fulfill a valid request

Depending on how the request was received, the status code and text is determined and sent back to the client to inform [24].

HTTP headers contain additional information about the response and the server that sent it. Typically this could be the time and date of the response message, the server ID of the server that responded, and Content-type which informs what kind of file was sent back.

The message body is an optional part of the response that can be utilized to give specific information back to the client. This could be a returned file from a successful request defined by the Content-type. The reason for message bodies being optional is that not all requests sent to a server expects a file in return. If a simple command is sent that only needs a status code and text (such as "204 No Content"), then a body is unnecessary and thus not added to the response [35].

2.7.2 Application Programming Interface

Application Programming Interfaces (APIs) are sets of predefined resources from software applications that allow other applications to access data and services without having to implement them themselves. The interface decides which data and services that are accessible and how to request the information [33].

APIs are built on request messages, authorization and response. A request message can be sent from the accessing application to the API, stating what service it would like to use. The API processes requests to determine what to do. The application offering an API usually requires an identification from the accessing application to register who is requesting its data. This is called an API authentication. API authentications are granted by the application offering the API and used by the accessing applications. If an application sends a request to an application API without authentication, the request is denied and a deny message is returned to the accessing application. If the authentication is accepted, the requested service or data is granted [46].

2.7.3 REST API

REST API (Representational State Transfer API) or RESTful API is an architectural style of how to construct APIs in a REST manner that was coined by Roy Fielding[14]. This style tries to reduce latency and network communications as well as increasing Independence and scalability of implementations [29].

In order to be RESTful, the API need to follow a set of design principles [25].

- A REST API should:
  - Use a uniform interface. All API requests for the same resource should follow the same design, regardless of requester.
  - Be client-server decoupling. The client and server must be independent from each other.
  - Be stateless. Each request should contain all necessary information to process it.
  - Resources should be cacheable on the client or server side.
  - Use a layered system architecture. Messages go through different layers and the API need to be designed accordingly.
  - Offer Code on Demand (Optional). In some cases a response message can contain executable code and should run on-demand.

2.8 Data collection

2.8.1 Brainstorming

Brainstorming is a creative thinking technique where the goal is to generate ideas, solutions and concepts. Multiple methods of brainstorming has been proposed and in this thesis, the brainwriting method was used.

Brainwriting is a method where the participants write down their ideas on a piece of paper during a set amount of time. The participants are encouraged to write down any idea they produce without considering its relevance or quality, in order to limit idea blockage. After the set time, the ideas written down are shared to the brainwriting group to be discussed and evaluated [40].

Since the author was alone on this project, the brainstorming session was completed in solitude.

2.8.2 Waterfall methodology

Waterfall methodology, or waterfall model, is a linear and sequential development approach that can be applied to software development. The name stems from that the development process "flows like a waterfall" through all phases of the project, with each phase completing before the next one begins. This thesis used this model in its design process.

The common stages of this method are:

- 1. Requirements

- 2. Design
- 3. Implementation
- 4. Verification or Testing
- 5. Deployment and Maintenance

The requirement phase gathers all information needed to get a detailed understanding of the project's goal and their requirements. This involves studying previous research, project and risk planning, and timeline scheduling.

The design phase introduces early development to structure a technical solution. This could be a low-level designed concept, or a high-level simulation to visualize and run functions and showcase what the software could potentially become.

Following the design phase, the implementation phase marks the start of implementing the software into the hardware. When the software is runnable, it can be tested to either progress to the next phase, or return to the design phase. In this project, the design and implementation phase was alternated in-between in a more trial-and-error procedure because of the author's limited area of expertise in wireless communication and android development.

Verification or testing with human subjects determines if the software is ready for deployment or not. This was the last step of the waterfall model that this project completed which resulted in usability tests with human subjects.

The deployment and maintenance phase is initiated once the software has been introduced to the market. Users may report problems along the way which will need to be supported. This thesis did not include this phase since the end goal was purely theoretical and not commercial [52].

2.8.3 Pilot study

A pilot study is a small-sample study made as a prelude to a larger scale study. The purpose is to collect data that can determine if the main study is feasible. It can also work as a guide for the main study to prevent errors or problems.

A pilot study can applied to user tests by running the test on a few test subjects to determine that the test works as intended. Here, potential problems can be caught and corrected to make the main test run smoothly and according to plan [10].

A pilot study was carried out as a preparation for the main user test of this thesis.

2.8.4 A/B Testing

A/B testing is an evaluation method used to compare two different versions of a design to conclude which one users may prefer. The A/B test introduces two variations of a service or product, version A and version B, into a simple controlled experiment where test users are randomly served a version to use in the experiment. The test users are divided into two equally large groups to form a complete sample size. One group test version A and the other test version B [56]. If the sample size is small, the test users are not divided into separate groups and instead test both versions. To negate any transfer-of-learning between A and B, the order of them must be changed in between test users [45].

A/B testing is a popular and proven data-collection method that give an indication on what version users prefer, but it has limitations. Because of its quantitative nature, A/B testing is most effective by combining it with qualitative evaluation methods to understand why test users prefer one version over the other.

2.8.5 NASA Task Load Index (TLX)

NASA Task Load Index is a multi-dimensional scale to estimate workload across six variables: Mental, Physical, and Temporal Demands, Frustration, Effort and Performance. The tool is used during or immediately after a task has been performed where the operator estimates the experienced workload on a scale ranging from LOW to HIGH. A total of 21 gradations are located on the scale to indicate a finer resolution to assist the operator with estimating the perceived workload more accurately. When the different variables have been estimated, a weighting scheme is used to produce an overall workload value for the performed task. The weighting scale was introduced to take individual differences into account when computing the overall workload. Personal weight factors is derived for each participant based on their perceived importance of each variable. However, the weighting process has been reported to actually decrease experimental validity. Therefore, the process was disregarded in the user tests [20].

2.8.6 System Usability Scale (SUS)

The System Usability Scale is a tool to estimate perceived usability of a system. The tool is made up of 10 statements that the user can agree or disagree with on a scale of 1-5 (1 being Strongly disagree and 5 being Strongly agree). When each statement has been scored, an overall score can be produced that can range from 0 (poor usability) to 100 (excellent usability). This score can give an overview of how subjects perceive the system. A score above 68 is considered to be a system above average. A score below 68 is considered below average. It is important that the tool is used immediately after a user has used the system to ensure accurate estimations [7].

2.8.7 Semi-structured interview

Semi-structured interview is an interview method to use when elaborated answers to a topic are of interest. Each interview is based around the same base questions but individual follow-up questions and expansions can arise for each interviewee to explore topics and paths that might not occur with other interviewees. In this way, each interview is likely to vary and gives space for unique answers and data to be collected [34].

Semi-structured interviews are effective to use when reason and opinion based answers are wanted. The interviewees are given space to think aloud and explain more in depth on why certain behaviour occurs and put their personal thoughts and experiences behind each answer. This can lead to deeper understanding of the research questions [34].

Chapter 3
Technical design

This chapter details the design of the project. The development process is described along with project directions, both debated and taken, early issues, and the overall project architecture is introduced. The chosen equipment is discussed and elaborated on and potential alternatives are debated and reasoned.

3.1 Choice of equipment

3.1.1 Control device

In the early stages of this project, the control device used to interact with IoT devices had to be chosen. Recent studies, such as the gaze-based AR research [3], suggest that AR technology can be used well to interact with devices in an environment if the AR device is designed in a manageable way and not be a hindrance to the user. Magic Leap, which were used in this research, and most other AR glasses available on today's market are quite large and bulky in design to be able to fit the desired hardware. This was a negative aspect that many users brought up in evaluations, explaining that Magic Leap were too exhausting to use in a relaxed setting such as a home. On top of this it felt redundant to plan a thesis on the same principles as previous research, so it was decided to move away from strict AR glasses to find a different device of lighter and smoother design.

The design of Google Glass seemed to be a better direction to explore in the regard of using a more manageable device. The mixed reality technique Google use in Glass is not as focused on delivering realistic content to the user, but instead focused on bringing clear information and assistance. The developers themselves describe it as Assisted Reality (aR) which is a bit different from Augmented Reality (AR) and seemed promising [13][16]. Glass also has built in sensors that could be used to explore IoT control in a similar was as the Ubi-compass [47], where rotating and aligning the wearable towards IoT devices let users discover and interact with them.

From a programming perspective it was of interest to explore an Android device since close to no experience was held in that area. Android offers an official documentation to guide a design process, and a small documentation for Glass that could help during development. Android also offer the Development environment Android Studio where apps and their graphical layout can be developed and tested on emulated devices.

Because of these reasons mainly, Google Glass was chosen as the control device for this project.

3.1.2 Wireless Communication

Next step was to map out communication and connect Glass to the IoT devices in a simple and effective way. One problem with using different smart devices from different developers is that they all use communication protocols and interfaces that potentially could differ from each other. If one device uses protocol 1 and user-interface A, and another device uses protocol 2 and user-interface B, the consumer would most likely need to use different smart phone apps to setup and control the two units. *"Commonly, appliances with any sort of intelligence and data collection originate from various vendors, rely on different connectivity standards and have different network interfaces. Yet, the concept of a smart home supposes every device and sensor can work together and this widely fragmented environment can consolidate into a unified system."* [28]. Naturally, by adding more smart devices from different brands into your smart network, the complexity of supervising them could potentially be a burden on the consumer. This issue can be reduced using Home Assistant.

By constantly expanding its number of integrations, Home Assistant can collect a large number of different smart devices under one interface and ease information exchange between devices and users. The Home Assistant interface can be accessed via a web browser or the Home Assistant app that is available for iOS and Android. The Home Assistant can also be reached by accessing its RESTful API to send and receive information from an application. Integrating this API communication into one application solves the issue of needing multiple different ones for different devices. Home Assistant is familiar with most of today's popular smart devices, so a user only need to understand the application communicating with Home Assistant in order to be able to control most devices.

The valuable abilities that Home Assistant posses made it a great fit to this project. Its API is well documented and it has both Sonos and Philips Hue controls as available integrations, saving valuable time in the design process. The saved time could instead be spent on focusing on communication to Home Assistant instead of having to spend time on developing specific communication for both Sonos and Hue separately.

Home Assistant was installed and ran on a Raspberry Pi which was connected wirelessly to a local router using a WiFi protocol. The Philips hue Bridge was connected to the network with an Ethernet cable, and Sonos was connected wirelessly to the same network in order for the Home Assistant to discover and add them to its pool of connected devices. This seemed like a natural home network setup. Glass would then be connected wirelessly to the same network and use it to send HTTP requests to the Home Assistant in order to control the smart devices and to get updated information about them.

3.1.3 Smart devices

When deciding on what kind of smart devices to use it was decided to use Philips Hue and Sonos Play:1. Both lights and speakers were regarded as central in a smart home environment and are common IoT brands on today's market. The devices were also available to borrow from the University which simplified the process. As stated before, both Sonos and Philips Hue have integrations in Home Assistant. It would save time to not having to implement specific communication between the user and these devices and instead fully trust Home Assistant to solve this. Furthermore, the devices are quite different in nature and fulfill different purposes for customers. Sonos is a speaker brand developing sound systems with music and sound interaction whereas Hue develop light systems where users can control light brightness and color. This would not only give more use cases for the application, but also bring another dimension into the final evaluation and results of the application and glasses.

Other possible devices that were considered were a Smart TV, network cameras, vacuum cleaners and home security systems. These were all deemed to fit the project due to their everyday house hold characteristics and possibility to be remotely controlled. The Smart TV was central during the development of the application but could not be brought to the university laboratory and was therefore dropped from the project in its later stages. The other ideas were abandoned along the way due to limitations of budget, time, implementation and other inconveniences.

3.2 Project Overview

3.3 Application development

During the initial stages of the project, time was spent on researching HTTP requests to get familiarised with the communication protocol. In order to get messages accepted by the Home Assistant API, a long-lived authorization token was needed. This was generated from a Home Assistant developer page and used in every HTTP request. Simple ping commands was the first goal to send from a computer to Home Assistant. When that was established a few goals were structured on what the application needed to be able to accomplish:

- Use HTTP method GET to receive information about Home Assistant entities.
- Use HTTP method POST to turn on or off Home Assistant entities.

These functions were regarded as the minimal requirements for the application to be considered able to interact with smart devices and for the project to meet. The aspiration for the project was to exceed these with further goals down the line that would be met if time was available.

After concluding how to use GET and POST in a correct manner using the command line of a computer, the base of device shifted from a computer to Google Glass. The development environment Android Studio was chosen since Glass are based on Android and used throughout the entire development process. A usable feature in Android Studio is to emulate devices such as smart phones to run an Android project in a virtual setup (see figure 3.1). Instead of a smart phone, a Google Glass profile was imported into the environment with same dimensions and pixel resolution as the physical glasses. This proved to be very helpful since it was faster to test the application virtually than having to having to manually upload the application to Glass every time.

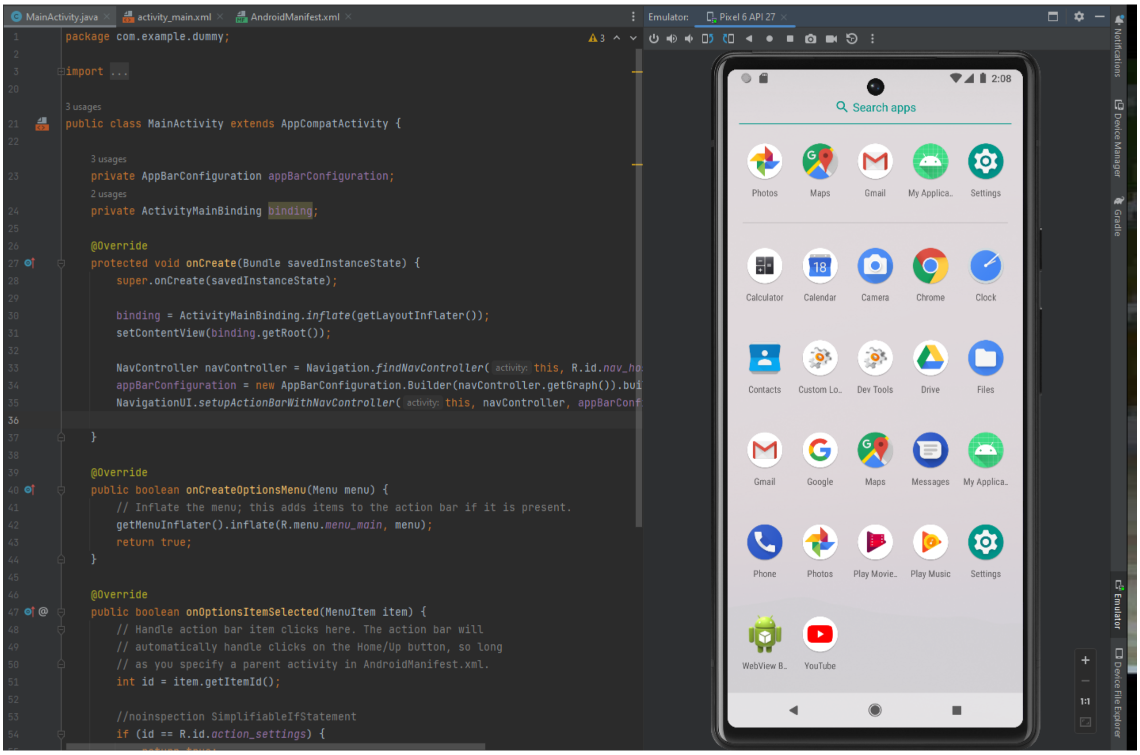


Figure 3.1: Screenshot of Android Studio and an emulated virtual smart phone

3.3.1 Project planning

Multiple ideas emerged from the brainstorming session and the early stages of the application interface development before any real structure had been carved out. Choices on graphical design, application navigation, interaction, orientation of user, HCI aspects such as visual indicators and more, needed to be taken in order to decide what path to follow for the application interface.

Jakob Nielsen's 10 Usability Heuristics is a set of principles for interaction design. These can be followed as guidelines while developing to prevent poor usability in a system [36]. These were studied with the aspiration to apply on the design to achieve positive results. These are:

1. **Visibility of system status** - The design should always keep users informed about what is going on, through appropriate feedback within a reasonable amount of time.

2. **Match between system and the real world** - The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon. Follow real-world conventions, making information appear in a natural and logical order.

3. **User control and freedom** - Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action without having to go through an extended process.

4. **Consistency and standards** - Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform and industry conventions.

5. **Error prevention** - Good error messages are important, but the best designs carefully prevent problems from occurring in the first place. Either eliminate error-prone conditions, or check for them and present users with a confirmation option before they commit to the action.

6. **Recognition rather than recall** - Minimize the user's memory load by making elements, actions, and options visible. The user should not have to remember information from one part of the interface to another. Information required to use the design (e.g. field labels or menu items) should be visible or easily retrievable when needed.

7. **Flexibility and efficiency of use** - Shortcuts — hidden from novice users — may speed up the interaction for the expert user such that the design can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. **Aesthetic and minimalist design** - Interfaces should not contain information which is irrelevant or rarely needed. Every extra unit of information in an interface competes with the relevant units of information and diminishes their relative visibility.

9. **Help users recognise, diagnose, and recover from errors** - Error messages should be expressed in plain language (no error codes) and should precisely indicate the problem as well as constructively suggest a solution.

10. **Help and documentation** - It's best if the system does not need any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.

The initial idea was to develop an application similar to a smart phone application in hopes that it would be familiar to users. This would include clickable buttons, scrollable menu's and a detailed dashboard offering plenty of options for the user to choose. Inspiration was taken from the official Home Assistant iOS tablet application which was later used in the user tests. However, after implementing and testing this layout, the author started questioning the approach and concluded that it was not as suited for Glass as it might be for smart phones and tablets.

While exploring the path heavily influenced by the iOS application layout, it was quickly discovered that it was not fit for Glass since it broke some of the usability guidelines stated above. While giving users plenty of options and information on screen might seem like a good idea at first, it can quickly escalate and burden the user. Trying to present information about every smart device available in an environment onto the Glass screen was deemed too problematic, since it broke guideline eight about minimalist design. The nature of Glass is to assist with valuable information and to not take up too much of the bearer's field of vision. This was a concern if the entire screen was crowded with too much information.

Another problem with the iOS inspiration was the accuracy of touch actions. The touch panel on Glass has an area of approximately 50x15mm, compared to a smart phone of 122x91mm touch screen [11]. This is circa 41% of the length touch direction and 17% of the wide touch direction and significantly smaller. This suggests that less accurate touch inputs can be given using Glass compared to a smart phone. Furthermore, a concern that if the same user would try to press a button on the same interface (scaled according to the smart device touch area), one on Glass and one on a smart phone, the user would have a harder time accurately giving the intended commands using the Glass touch pad compared to a smart device touch screen.

3.3.2 Navigation

Without implementing any touch control routines, Glass used its default navigation which highlight the first interactable object on the screen. Objects, such as buttons, drop-down menus and seek bars to name a few, can be selected by swiping backwards and forwards and activated by a tap motion (see figure 3.2 for early stage development test). A problem with this was that the swipe motions were exclusive for selecting objects and could not be used for anything else without implementing custom motion controls. The default navigation system was abandoned for this reason.

Instead, the touch controls were defined by implementing a *gestureDetector* class that would constantly listen for inputs on the touch pad. When an input was registered, the gesture-Detector processed it and executed a navigation command specified by the developer. The gestures available to the user was determined to: One finger swipe up, down, forward and backwards, two finger swipe up, down, forward and backwards, and one finger tap. One finger tap and swipes were reserved for smart device interaction and two finger swipes were reserved for navigating the application interface with emphasis on the 4th guideline about

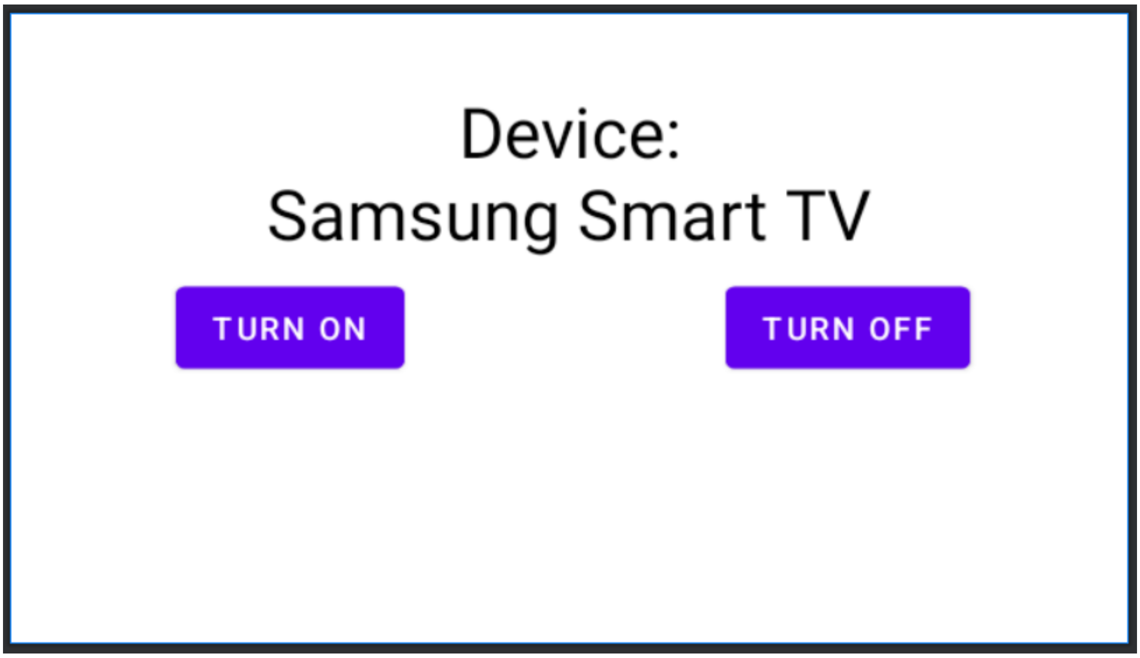


Figure 3.2: Early stage development using two clickable buttons

consistency and standards.

Choosing two fingers as navigation was a risk the author was aware of and took some time to decide. When studying sample application and demos provided by Google, most applications were navigated by using one finger swipes [18]. This is also how you navigate the Glass operative system and changing it was a risk since it violates the guideline of consistency. Users would probably not use two fingers when using the application at first but hopes were that they would be able to learn and accept it. This would contribute to guideline 7 about flexibility and efficiency of use, since these two finger swipe shortcuts could lead to a smoother application for an experienced user. One of the main functions, and the main way to circle through devices, were to discover devices by looking at them. This would make the touch controls used to swap between devices less of a priority. It was also of interest to evaluate an alternative way of navigation so the two finger touch navigation was kept.

3.3.3 Navigation: Discovery mode

A central idea for the project was to let users discover smart devices by glancing across the environment. A similar idea as the UbiCompass was explored where Glass would be able to detect devices located in front of them. This would let users know which devices they can interact with by simply looking at them. By implementing a *sensorManager*, a class provided by Androids standard library, the Glass' sensors can be accessed. The *sensorManager* provide two methods, *getRotationMatrixFromVector* and *getOrientation*, which are used to calculate the glasses azimuth, or current horizontal angle relative to the earth's magnetic field. Fetching a rotation vector provided by the Glass sensors and sending it into *getRotationMatrixFromVector* constructs a rotation matrix. This matrix is then processed by *getOrientation* to compute the current heading.

The sensors proved to be quite difficult to work with and was not giving precise numbers. Roll, Pitch and Yaw are three measurements, besides the azimuth, that could be fetched from the sensors (see figure 3.3) and gave some interesting insights while developing. The Pitch measures 90 at horizontal level and if this value was too low, e.g the glasses tilt too far up or down, it would impact the azimuth and making it inaccurate. Roll also had a negative impact on the orientation accuracy which could limit the user when trying to discover a smart device when looking towards it.

Multiple measures were made to observe what impact Pitch had on the azimuth (see table 3.1.

Table 3.1: Azimuth measures for different fixated pitches

Pitch	Azimuth: Philips Hue				Azimuth: Sonos			
80	329	8	349	4	283	310	309	314
85	353	5	350	353	300	298	305	302
90	102	107	90	43	185	152	198	183

These inaccuracies were big enough to not be able to fixate the devices location into the application but they were small enough to work around.

Despite the inaccuracy of the sensors it was mostly towards earths magnetic field and still showed potential to function well for a discovery mode. If the smart devices can be found relative to the glasses, then showing inaccurate measures of where earth's north pole is located is of no concern.

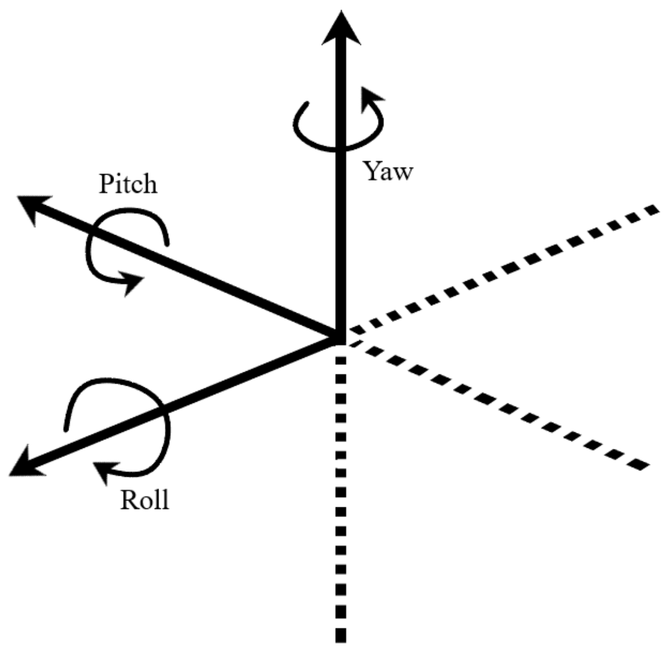


Figure 3.3: Illustration of Pitch, Yaw and Roll

3.3.4 Interaction

When developing the interactive aspect of the application, emphasis on both hardware and software were taken in consideration to try and offer a user friendly experience. The decision to restrict one finger touch inputs for control commands when interacting with smart devices was taken based on the 4th guideline about usability. These were set as following:

- **TAP** - On/Off
- **SWIPE UP** - Increase
- **SWIPE DOWN** - Decrease
- **SWIPE FORWARD** - Next
- **SWIPE BACKWARD** - Previous

The idea to use these were that even if a device does not find all these commands useful, it can at least find one or more of them useful. For one-dimensional devices, the **TAP** control could be used to switch them on or off but for more complex devices such as Philips Hue and Sonos, most touch inputs could be utilized.

The main interactions for the Philips Hue light bulb was determined to be the following:

- Turning the light bulb on and off
- Increase and decrease brightness of the bulb
  - **Optional interaction:** Change color

The touch input **TAP** was naturally added to the On/Off interaction. If the light bulb was currently turned on, the **TAP** input would turn the light off. If the light bulb was currently turned off, **TAP** would turn the lights on.

When considering which touch input to bind brightness increase/decrease to there were two set of options that seemed most intuitive.

Table 3.2: Touch gestures up and down

Touch input	Action
SWIPE UP	Increase brightness
SWIPE DOWN	Decrease brightness

The set of inputs stated in table 3.2 and table 3.3 both seemed viable and usable. To switch the order of increasing and decreasing (e.g Up or Forward to decrease and Down or Backward to increase) was not regarded as intuitive by the author and therefore not considered. The first set of commands were ultimately chosen after considering the touch inputs for interaction

Table 3.3: Touch gestures forward and backward

Touch input	Action
SWIPE FORWARD	Increase brightness
SWIPE BACKWARD	Decrease brightness

with the Sonos speaker.

The main interactions for the Sonos Play:1 speaker was determined to be the following:

- Starting and stopping a song
- Increasing and decreasing volume
- Skip to the next or previous song

The **TAP** command was again chosen as the toggle command, this time to start and stop a song. This seemed intuitive and similar to the light bulb On/Off command. It also follow guideline 4 of consistency and standard. When the **TAP** input was given via the touch pad, a paused song would start from its current timestamp. If the **TAP** input was given for a song that was currently playing, it would pause on its current timestamp.

When deciding on the swipe gestures for increasing and decreasing volume it was again a question of which direction to swipe on the touch panel. Both directions were valid sets that could fit this function and when tested during development none stood out more than the other. How to go forward to the next song or backwards to the previous song however seemed like a natural fit for forward and backward swipe and not as fit for the vertical swipe directions. It was decided to use **SWIPE UP** and **SWIPE DOWN** to increase and decrease volume, and **SWIPE FORWARD** and **SWIPE BACKWARDS** to go to the next or previous song. This impacted the choice of touch gestures for the Philips Hue light bulb and cemented the idea to use vertical swipe directions universally across the smart devices to increase or decrease.

Chapter 4
User Test

This chapter introduces the user test environments, structure, preparation, execution and the participating test subjects. It describes how the test environment was set up in the lab used at Lund University throughout the project. The test scenario structure is described and the usage of A/B test and different data collection tools is stated. The execution of the user tests is described to inform how a test looked. The test subjects background information is presented in the closing part of this chapter.

4.1 Environment

The user tests were carried out in a lab environment at Lund University (see figure 4.1 and 4.2). The lab was equipped with cameras and microphones to collect video and sound material from each test. The smart devices were placed on opposite sides of a table with the test subject sitting in between them on the far side if the table. This gave the test user enough space to look at each device without having them both in their field of vision at any time. A microphone placed in the middle of the table was facing the test subject to capture sound. A camera facing the subjects right side were used to capture the subjects movements and expressions as well as the touch pad where all swipe and click commands were given to the application. The captured sound and video were later examined to analyze the subjects behaviour during experiment.

4.2 Structure

Using an iPad and Google Glass, the test subject were to follow a set of instructions given by the user test supervisor to complete a scenario about interacting with the smart devices. The same scenario was carried out by a subject twice, once using Google Glass (test A) and once using the iPad (test B). The subjects were asked to talk throughout the scenarios and explain their thoughts, expectations and actions. This would prove to give a lot of valuable feedback

Figure 4.1: User test setup

for future implementations. After each scenario was completed, a semi structured interview was conducted where the subject got to describe more in detail what they thought of the scenario, glasses and application. This was followed by a NASA TLX and SUS evaluation which were analyzed together with the video and sound material.

The test scenario was constructed in a certain order to make the test subject interact with each smart device in different ways. A fictional scenario was described where the subjects were asked to imagine themselves in their living room using either Glass (test A) or the tablet (test B) connected to their wireless home network. Using the control device, the test subjects were instructed to turn on the lights, change brightness, turn on music, alter the volume level and finalized by turning off the lights and pausing the music.

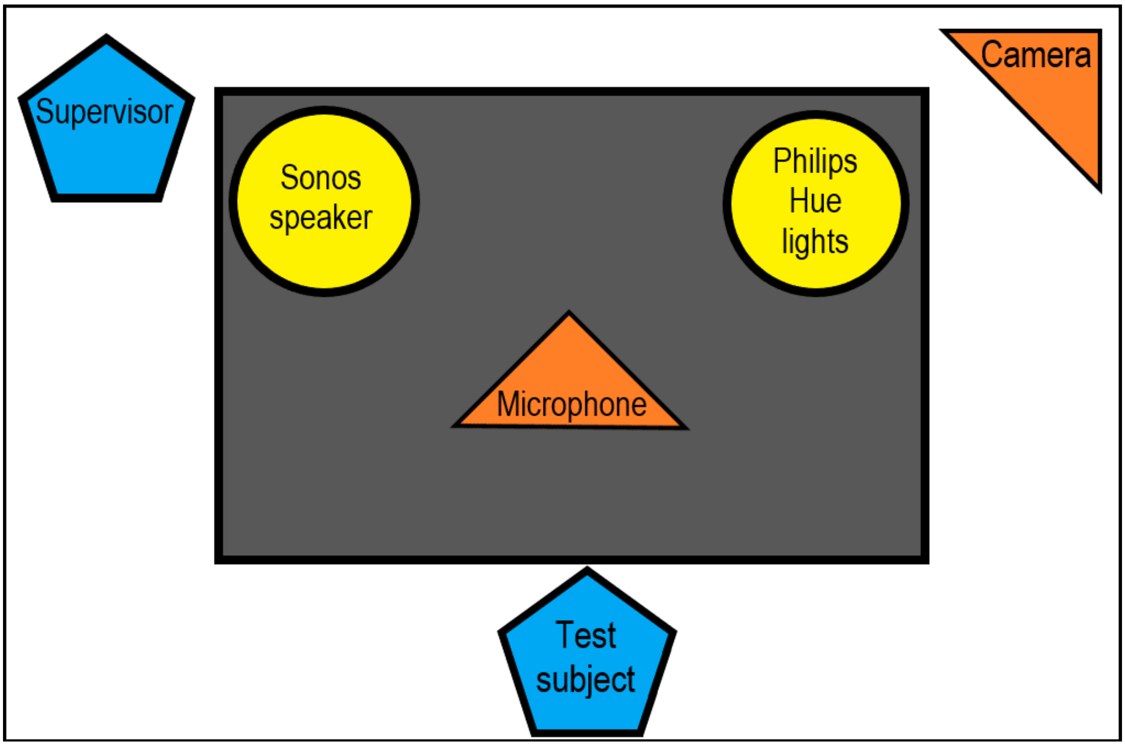


Figure 4.2: Graphical illustration of the test setup

For test scenario A and using Google Glass, subjects were asked to:

- Open the application
- Enter the discovery mode and calibrate the orientation
- Discover and interact with the Philips Hue light bulb
- Give a brief status update and describe the current state of the smart device
- Turn on the lights
- Increase and decrease the brightness
- Discover and interact with the Sonos speaker
- Give a brief status update and describe the current state of the smart device
- Play music on the speaker
- Increase and decrease the volume
- Change the current song playing
- Give a brief status update and describe the current state of the smart device
- Pause the music and turn off the light
- Close the application

For test scenario B and using an iPad, subjects were asked to:

- Give a brief status update and describe what current states their smart devices were in
- Turn on the Philips Hue light
- Increase and decrease the brightness
- Play music on the Sonos speaker
- Increase and decrease volume
- Change the current song playing
- Give a brief status update and describe what current states their smart devices were in
- Pause the music and turn off the light
- Close the application

The scenario for test A was modified slightly to make room for the discovery mode developed for Google Glass. The order of which the tests were carried out was distributed in different orders equally over the set of test subjects. This was done in order to observe if users starting with Glass (test A) and finishing with iPad (test B) would perform different than users starting with iPad (test B) and finishing with Glass (test A).

4.3 Preparation

An informative survey was filled out by each test subject preceding the experiment. Data such as age, technical interest and prior knowledge was of interest to analyze correlation between personal background and the results. They were also asked about their favorite musician and if they would like coffee or tea during the test. The intention for this was to give a more relaxing impression on the subjects and to remove any expectations of stress or performance foundation for the test. The musician they gave as an answer were later loaded into the speaker software to play when the subject interacted with it. The purpose for this was to bring in something familiar to the experiment in order to simulate a home environment they could relate to.

When the test subject arrived to the lab they were seated in the appointed test chair and given a brief introduction to the thesis, the test they would be participating in and its purpose. Following this was a description of the test structure, what and how data was being collected as well as a consent form to fill out. The consent form informed them of their rights, how the data would be stored and used, and how they could back out or have the data destroyed at any time prior of the thesis being presented and published.

If a test subject was unfamiliar with the device they would use for the scenario, they were asked to test and explore it freely, excluding the control application they would use later.

The intention was to make each subject comfortable navigating the new device so that each test could be completed. When the subject felt comfortable the test could begin.

4.4 Execution

The main purpose of the supervisor during the experiment was to observe and manage the scenario. Minimal influence on the test was intended and except for giving instructions on what task to complete, the supervisor assisted only if the subject could not manage to proceed for a longer period of time. The subjects were asked to continuously describe their thought process, expectations and actions to get a deeper understanding of their personal opinion, previous knowledge and experience as well as surprises and unexpected outcomes. If something did not go according to the subjects expectations, it was later brought up by the supervisor in the interview to be discussed.

After the scenario was completed the semi-structured interview followed to evaluate the subjects impression of it and to follow up any interesting occurrences. The interview followed a set of questions regarding their opinion on the application, control unit and overall feelings toward smart glasses and IoT solutions. The question were both specific and open ended to give room for the subject to express thoughts freely. Some individual questions were added during the test to explore topics that specifically occurred for that person.

Finally after the semi-structured interview, both the NASA TLX and SUS forms were filled out to get an overall quantitative evaluation of the test.

4.5 Participants

Thirteen test subjects showed interest in participating. In total, twelve subjects appeared for their scheduled time and all of them completed their test. The subject age ranged from 23 to 63 with a mean age of 34 (see figure 4.3) and consisted of eight males and four females (see figure 4.4) across different educations (see figure 4.5) and professions (see figure 4.6). Most subjects estimated their own technical level as four on a 1-5 scale (see figure 4.7). Every subject was familiar with the concept of Augmented Reality and Virtual Reality. Six subjects stated that they had previous experience with AR and VR, and six subjects had no prior experience. All subjects had experience with IoT devices and owned some form of IoT device. Four subjects does not use glasses, four subject use glasses and four subjects use occasionally.

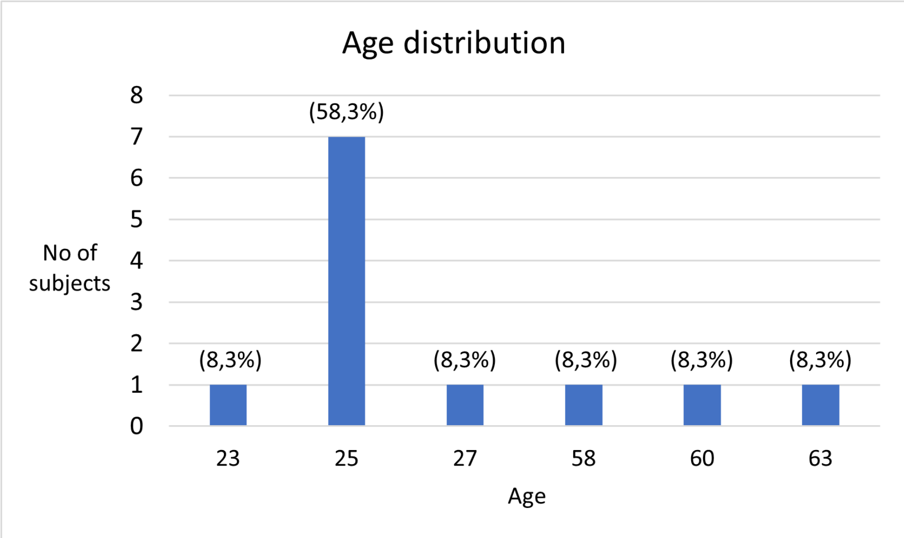


Figure 4.3: Age distribution of test subjects

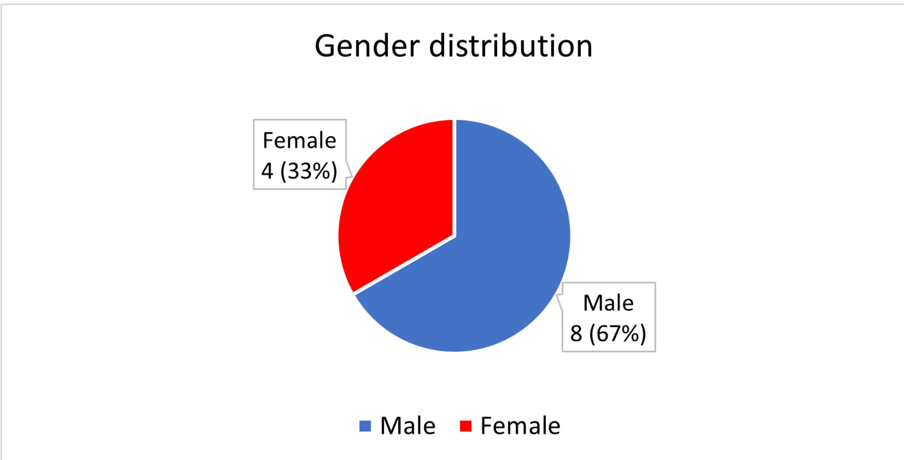


Figure 4.4: Gender distribution of test subjects

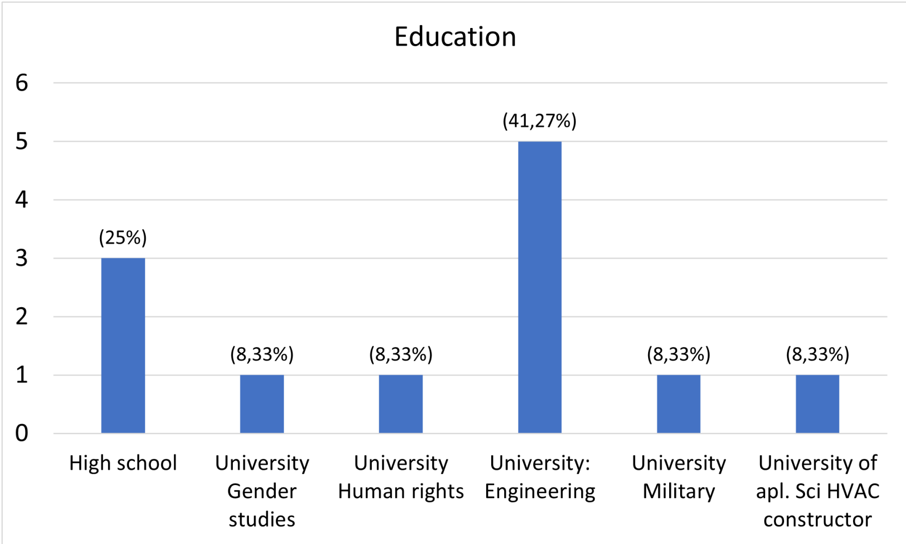


Figure 4.5: Education distribution of test subjects

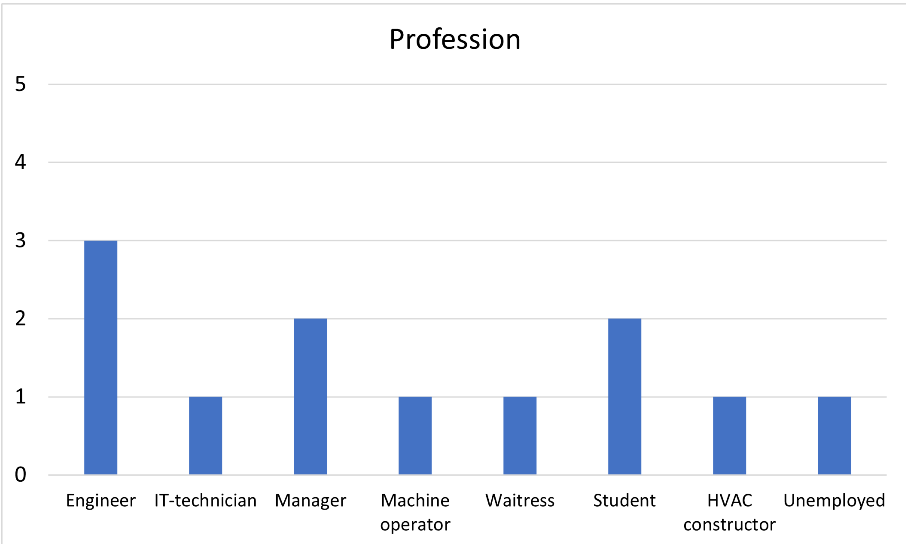


Figure 4.6: Distribution of professions

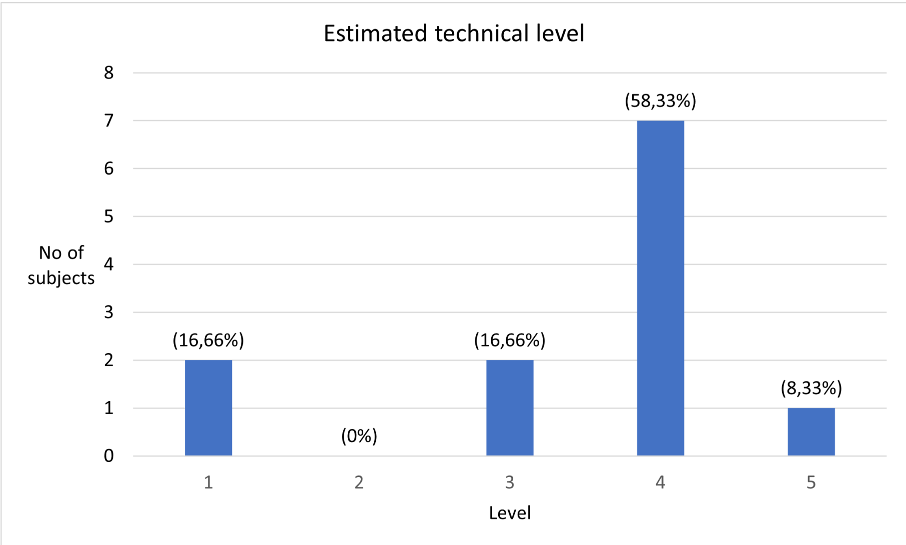


Figure 4.7: Estimated technical level

Chapter 5
Results

This chapter presents the results collected from the user tests. It describes finished application aspects, such as the graphical interface, navigation, communication, as well as functionality. This is followed by the quantitative results from the user tests such as performance, usability and task load estimation.

5.1 Application

5.1.1 Graphical Interface

The user interface took on a minimalist style to not overwhelm the user with information. Whenever a command was sent, a small notification would appear on the bottom side to inform a user which command was registered by the touch pad (see figure 5.3). Three layouts were designed, one for Philips Hue, one for Sonos, and one for the discovery mode.

The layout for Philips Hue consisted of four text fields and two indicators. The top center text field informed the user of what current device they were controlling, which, for this device, stated "Philips Hue Lights". The middle text field showed what current state the lights were in. This could either be "ON" or "OFF". This text was accompanied with a background indicator to enhance the information by turning green for "ON" and red for "OFF". On the far right, a bar illustrating the current brightness level was inserted to give a visual aid to the text field showing the numerical brightness. The numerical brightness ranged from 0-255 which was the same range as the Philips Hue API provided. This might not be intuitive for all users but this was kept intentionally to observe if any test subjects noticed and raised any opinions about it. The color for current brightness level was chosen to be yellow to illustrate light. Lastly, a text field "Brightness" was included to inform the user that the bar and number describes current Brightness (see figure 5.1 for the complete layout).

Figure 5.1: User interface when interacting with Philips Hue

The Sonos layout followed a similar style as the Philips Hue layout. A text field "Sonos WiFi Speaker" indicates which smart device to control from this view. A current status text with an underlying color indicator, informing if the music is "PLAYING" or "PAUSED". Lastly, an indication bar illustrating the volume level and number (see figure 5.2) was placed on the right side. Both the status text and indicator was only visable when a song was loaded. If the speaker had no song loaded they would be hidden from view (see figure 5.3). No supportive text was added here to explain what the bar and number was indicating. This too was intentional to observe if any test subjects took notice of it and raised a preferable layout. The numerical indicator ranged from 0-100 which was the same as Sonos API provided, which could be more intuitive than the brightness range in Philips Hue, and was kept to observe any user opinions on the topic. The color for current volume level was set to green. A unique addition to the Sonos layout was the text field "Now playing: " which informed the user of what current song was loaded. This was updated concurrently with Sonos current playing song.

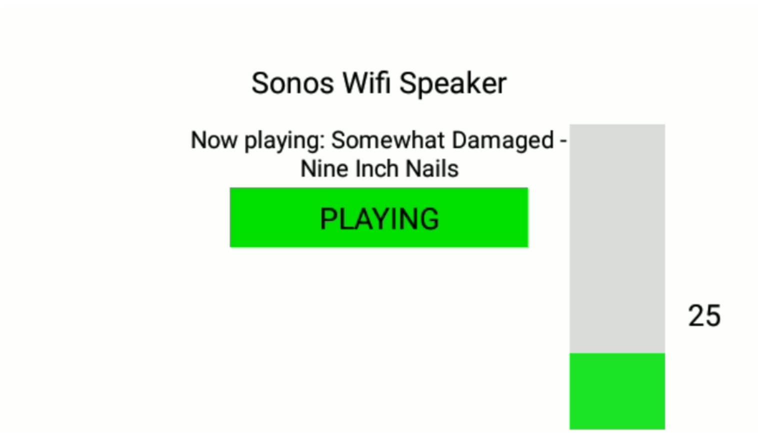


Figure 5.2: User interface when interacting with Sonos Play:1

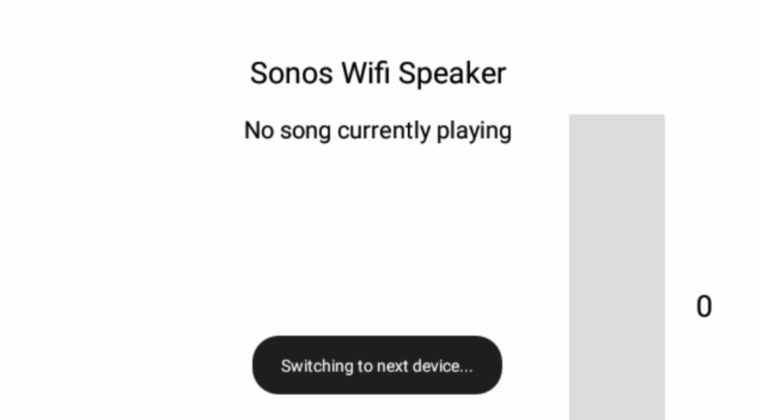


Figure 5.3: User interface when switching device

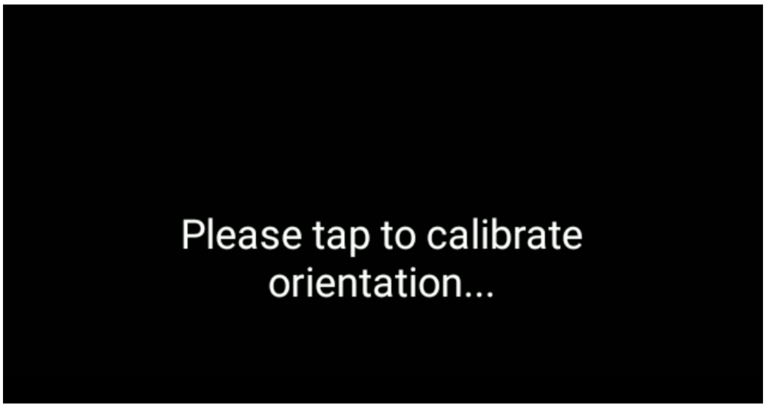


Figure 5.4: User interface for discovery mode: not calibrated

When a user entered the discovery mode for the first time they were instructed to calibrate their orientation. They were presented with the instruction page (see figure 5.4) until they proceeded with the calibration. When calibrated, they would be met with the calibrated layout (see figure 5.5) which would be present until the user either started a re-calibration, glanced over towards an interactable smart device, entered a smart device or closed the application (see figures 5.4,5.6 and 5.7).

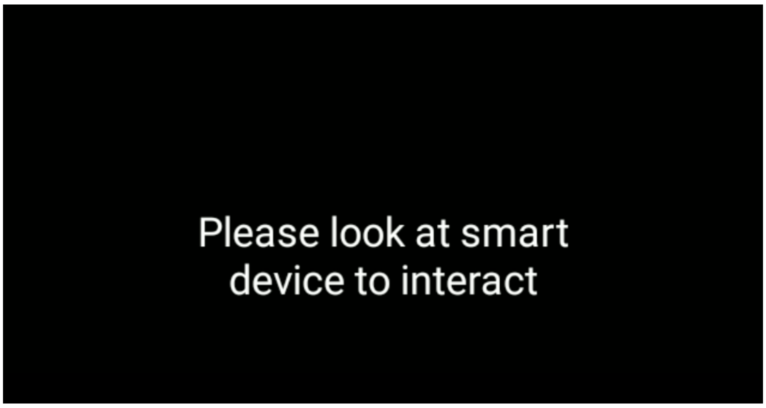


Figure 5.5: User interface for discovery mode: calibrated

Figure 5.6: User interface for discovery mode: finding Philips Hue

Figure 5.7: User interface for discovery mode: finding Sonos

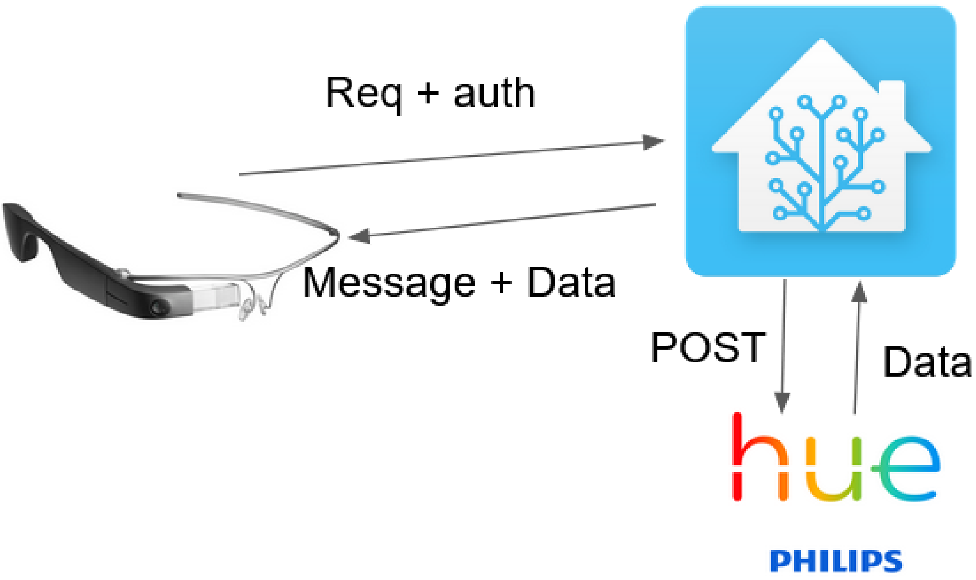


Figure 5.8: Illustration of communication structure

5.1.2 Communication structure

Communication between the user and the smart devices followed a simple structure (see figure 5.8). Touch gestures from the user to Glass' touch pad were processed and sent to the application. Inputs reserved for smart device control constructed and sent an HTTP request, containing all the necessary components, via the local wireless network to the Home Assistant software. Home Assistant would then process the request and act accordingly:

- GET request - Home Assistant fetches the specified entity's information which is stored locally on the Home Assistant. This information is sent back to Glass which is displayed on the application's graphical interface.
- POST request - Home Assistant sends the command to the specified entity and updates the requested state. Home Assistant then updates its own local information about the entity and sends this back to Glass, which is displayed on the application's graphical interface.

5.2 User Tests

The user tests described in chapter 4 took place in the Usability Lab at Lund Technical University.

A pilot study was carried out using the first two subjects to make sure the user tests functioned as expected and could be completed without issues. The pilot study yielded valuable results which showed that the application developed was not fully ready to be tested. Changes had to be made to how the Glass orientation was obtained and calculated. This was improved and updated for the rest of the test subjects. Because of the faulty software, the results from these subjects could be impacted negatively which the author realize. However, because of the small amount of test subjects in the main user test, the results from the pilot study participants are still used in the overall result.

5.2.1 A/B test

The A/B test was set up as explained in the previous chapter. There was no set distribution of which order test A and B was carried out between each subject. Five out of the twelve subjects started with scenario B, and seven out of twelve started with A, making it a 42% to 58% ratio. The pilot test was seen as the most problematic one and could be disregarded in the evaluations because of its nature, making the distribution 45% to 55% ratio.

Most test subjects completed both test A and B and most were vocal during the whole scenario. The WiFi used for the experiment was located in the basement of the building and provided varying strengths of signal between tests and subjects making it inconsistent. The result of this was longer response times for both applications used in test A and B and could have had a potential negative impact on the subject's evaluation due to inconsistency, irritation and confusion.

Test A

Test A introduced the test subjects to Google Glass and for most users, this was their first time using a pair. A short on-boarding procedure was held for each subject until they claimed to be ready for the test scenario. The application used was developed by the author and can be seen in figures 5.1 to 5.7.

Six out of the twelve subjects managed to use the discovery mode successfully as intended.

Three out of the twelve subjects used the discovery mode to somewhat success. One of them could find the indicator for Philips Hue at multiple locations besides the intended one. This user also needed to re-calibrate several times in order to find the Sonos speaker at the intended location. Another user found the Sonos indicator when looking too far past the physical speaker and not at the intended location. The last user managed to find and interact with Philips Hue after several tries but were unsuccessful on discovering Sonos.

Three out of the twelve subjects failed to use the discovery mode successfully. After calibrating several times, they were unable to discover any of the smart devices and instead forced to manually swap between the smart devices, using two finger swipes, to finish the test.

In total: 50% managed to use the discovery mode successfully, 25% managed to use it somewhat successfully and 25% were unsuccessful (see table 5.1 for complete discovery mode results from test participants).

Table 5.1: Discovery mode results

Test subject no	Discovered Philips Hue	Discovered Sonos
1	Yes but at different locations	Yes after several tries
2	Yes after several tries	No
3	No	No
4	No	No
5	No	No
6	Yes	Yes
7	Yes	Yes
8	Yes	Yes but slightly far off
9	Yes	Yes
10	Yes	Yes
11	Yes	Yes
12	Yes	Yes

Table 5.2: Philip Hue interaction results

Test subject no	Turn on light	Turn off light	Increase and decrease brightness
1	Yes	Yes	Yes - after swiping wrong direction at first
2	Yes	Yes	Yes
3	Yes	Yes	Yes - after swiping wrong direction at first
4	Yes	Yes	Yes - after swiping wrong direction at first
5	Yes	Yes	Yes - after swiping wrong direction at first
6	Yes	Yes	Yes - with a hint from supervisor
7	Yes	Yes	Yes
8	Yes	Yes	Yes
9	Yes	Yes	Yes - after swiping wrong direction at first
10	Yes	Yes	Yes
11	Yes	Yes	Yes
12	Yes	Yes	Yes

When sending commands to the smart devices some different intuitive touch inputs could be observed between the subjects.

When interacting with Philips Hue, each subject had the same idea on how to turn on the light bulb and did so by using input command **TAP**. This was the correct input which all subjects completed. When tasked to increase and decrease brightness however, six subjects first tried **SWIPE FORWARD** and **SWIPE BACKWARD** instead of the correct commands **SWIPE UP** and **SWIPE DOWN**. For the six subjects who did not get it on the first try, five subjects figured it out on their own and one needed guidance from the supervisor.

In total: All of the subjects managed to turn on and turn off the lights. Six of the subjects managed to increase and decrease brightness level on the first try, 42% managed to do it after several tries and 8% after guidance. How many tries it took to get it right was neglected as all number of tries was regarded as the same as long as the user managed to get it without help (see table 5.2 for complete Philips Hue control results from test participants).

Table 5.3: Sonos interaction results

Test subject no	Start and stop music	Increase and decrease volume	Change current song
1	Yes	Yes	Yes
2	No	No	No
3	Yes	Yes	Yes
4	Yes	Yes	Yes
5	Yes	Yes	Yes
6	Yes	Yes	Yes
7	Yes	Yes	Yes
8	Yes	Yes	Yes
9	Yes	Yes	Yes
10	Yes	Yes	Yes
11	Yes	Yes	Yes
12	Yes	Yes	Yes

When controlling the Sonos speaker, all but one managed to complete the tasks all on their own. Some unfortunate events such as application bugs and crashes occurred for three subjects during test A. For two subjects the incidents were minor and they were able to finish the test after restarting the application. Unfortunately, for subject two, the application kept on crashing and was not able to complete the Sonos interaction.

Test B

Test B introduced the test subjects to an iPad. Every user had previous experience using one thus making an on-boarding process unnecessary. The application used was developed by official Home Assistant developers and can be seen in figure 5.9. The discovery mode was not featured for the iPad and followed a simpler structure.

All test subjects managed to complete the test scenario to the end. When interacting with Philips Hue, five test subjects used the intended On/Off button to turn on the lights which is the intended way. Seven subjects instead turned on the lights by using the brightness bar. Since this is a possible way to complete the task it was regarded as correct. When tasked to turn off the lights, all subjects who previously used the brightness bar now found the On/Off button after discovering that turning the brightness bar all the way down put the brightness on 1% instead of off. Four out of the five subjects who previously used the On/Off button to turn on the lights used it to turn off the lights. One user who previously used the On/Off button to turn on the lights had done so by mistake and went for the brightness bar only to come to the same conclusion as the others (see table 5.4 for complete discovery mode results from test participants).

When interacting with the Sonos speaker, all test subjects managed to start and stop a song as well as change the current song playing. This was completed by using the media control symbols located on the interface. Pressing the play icon started a song from its current timestamp, and pressing pause stopped it on its current timestamp. Pressing the forward

Table 5.4: Philip Hue interaction results

Test subject no	Turn on light	Turn off light	Increase and decrease brightness
1	Yes - using the slider	Yes - after several tries	Yes
2	Yes	Yes	Yes
3	Yes - using the slider	Yes - after several tries	Yes
4	Yes - using the slider	Yes - after several tries	Yes
5	Yes - using the slider	Yes - with guidance from supervisor	Yes
6	Yes - using the slider	Yes - after several tries	Yes
7	Yes - using the slider	Yes - after several tries	Yes
8	Yes - using the slider	Yes - after several tries	Yes
9	Yes	Yes - after several tries	Yes
10	Yes	Yes	Yes
11	Yes	Yes	Yes
12	Yes	Yes	Yes

or backward icon skipped to the next song or to the previous. Increasing or decreasing the volume proved to be quite challenging. The volume bar had to be located by accessing the entity's advanced settings. This was accomplished by pressing the options icon up in the entity's top right corner. Three out of the twelve subjects managed to increase and decrease the volume successfully. Six out of the twelve managed after several tries were made in different ways. Three out of the twelve needed guidance to complete this task (see table 5.5 for complete discovery mode results from test participants).

Table 5.5: Sonos interaction results

Test subject no	Start and stop music	Increase and decrease volume	Change current song
1	Yes	Yes - after several tries	Yes
2	Yes	Yes	Yes
3	Yes	Yes - after several tries	Yes
4	Yes	Yes - after several tries	Yes
5	Yes	Yes	Yes
6	Yes	Yes - after guidance	Yes
7	Yes	Yes - after several tries	Yes
8	Yes	Yes - after several tries	Yes
9	Yes	Yes	Yes
10	Yes	Yes - after guidance	Yes
11	Yes	Yes - after guidance	Yes
12	Yes	Yes - after several tries	Yes

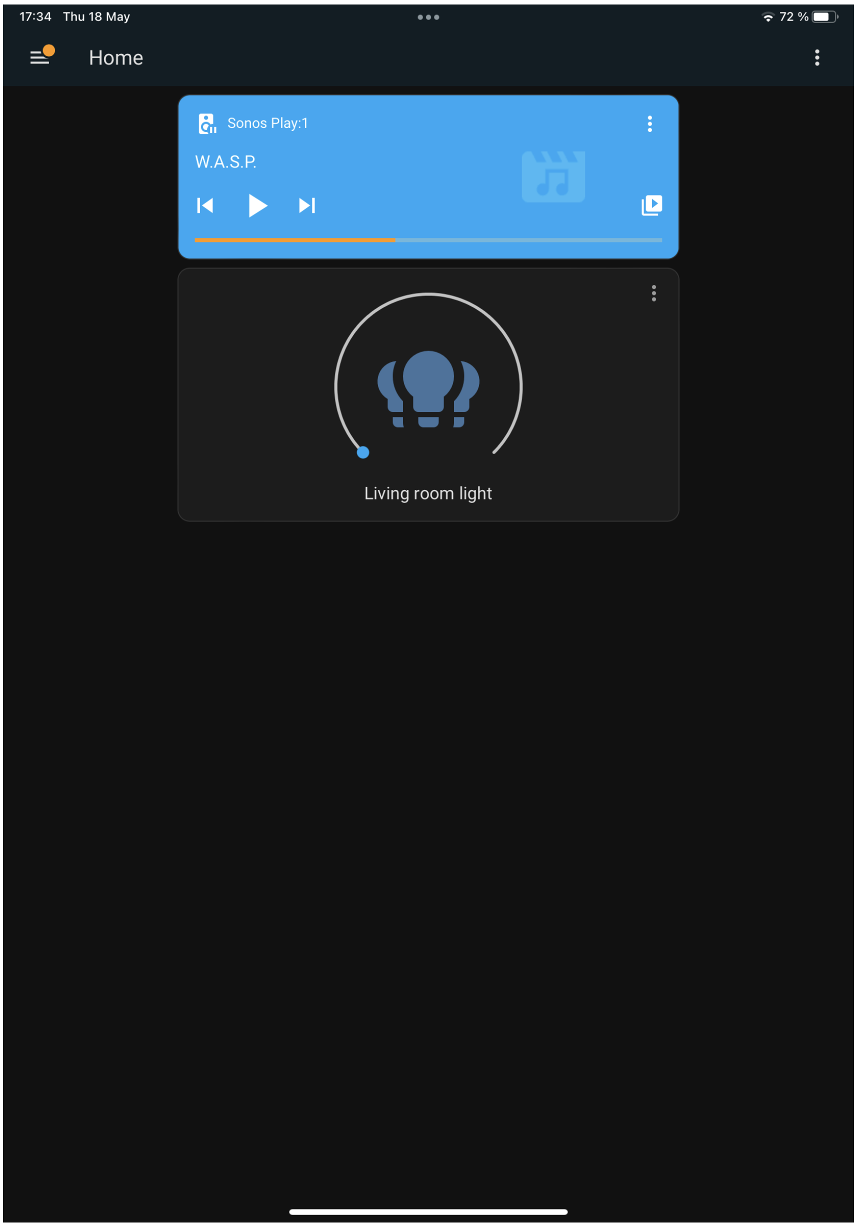


Figure 5.9: Home Assistant on iPad used during tests

5.2.2 Interview

The semi-structured interview that followed each test consisted of a set structure of base questions that each subject was given. Some questions were added or removed, depending on which control device the subjects just had been using during the test. Further questions were also added depending on the test subject and what happened during the test.

- How did you think the test went?
- Was it something that kept you from completing the test in a smooth manner?
- Did the navigation of the app feel natural or was something not as expected?
- What did you think about the touch controls? Did swipe and tap feel natural?

- Was there any problematic function that you would like to change or remove?
- Was there anything in the graphical interface that was misleading or illustrated something you did not expect it to do?
- Is there any function you would like to include in the application?
- How do you feel when you interact with IoT devices this way?
- Do you feel like you would rather interact physically with the IoT devices instead of using this solution?

After a subject had tested Google Glass, the following questions were added to the interview:

- How do you feel when you are wearing these smart glasses?
- Would you use them in your every day life?
- Would you be more comfortable wearing them if they were designed to look more like regular glasses? How big of an impact does this have on the wireless control system you just tried?
- How did you feel about the discovery mode? Can you see any pros and cons about using it in an environment equipped with IoT and smart devices?
- Would you be comfortable using other types of input controls such as voice command? Can you think of any other way?
- Can you imagine a future where you use a pair of smart glasses to interact with things in your environment?

When the subject had finished both tests, these questions were added to the final interview:

- What device do you prefer: the iPad or Google Glass? Why?
- Do Glass have any strengths that the iPad does not have?
- Do the iPad have any strengths that Glass does not have?

5.2.3 NASA Task Load Index

Overall average task load index for iPad: 29.77
Overall average task load index for Glass: 42.27.

Difference between Glass' and iPad's average task load index: 42% (rounded to nearest integer). Complete result table in figure 5.10.

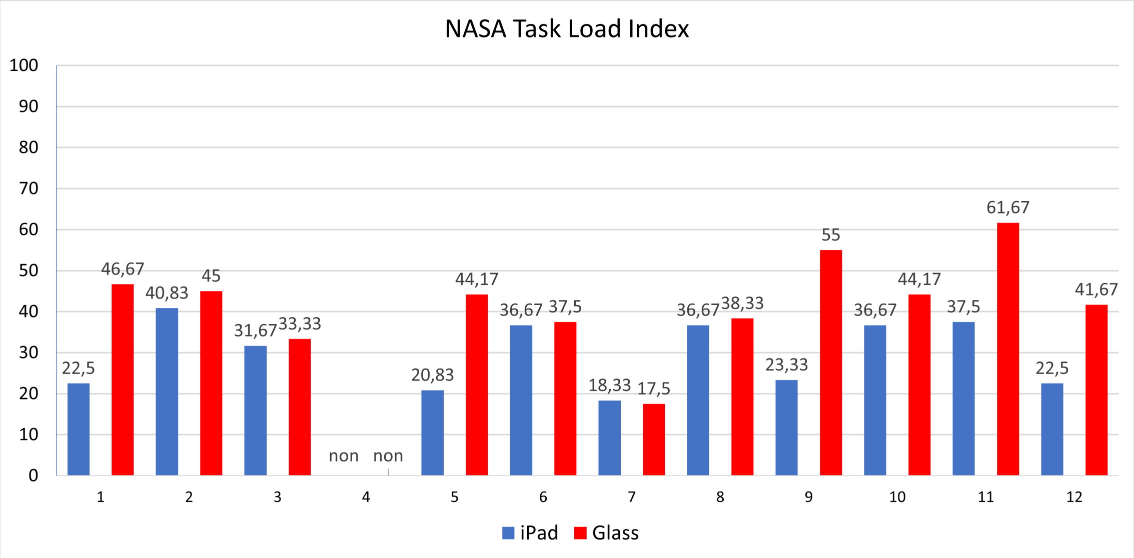


Figure 5.10: Bar diagram of NASA Task Load Index score

5.2.4 SUS evaluation

Overall average SUS for iPad: 90.42
Overall average SUS for Glass: 67.08

Difference between Glass' and iPad's average System Usability Score: 35% (rounded to nearest integer). Complete result table in figure 5.11.

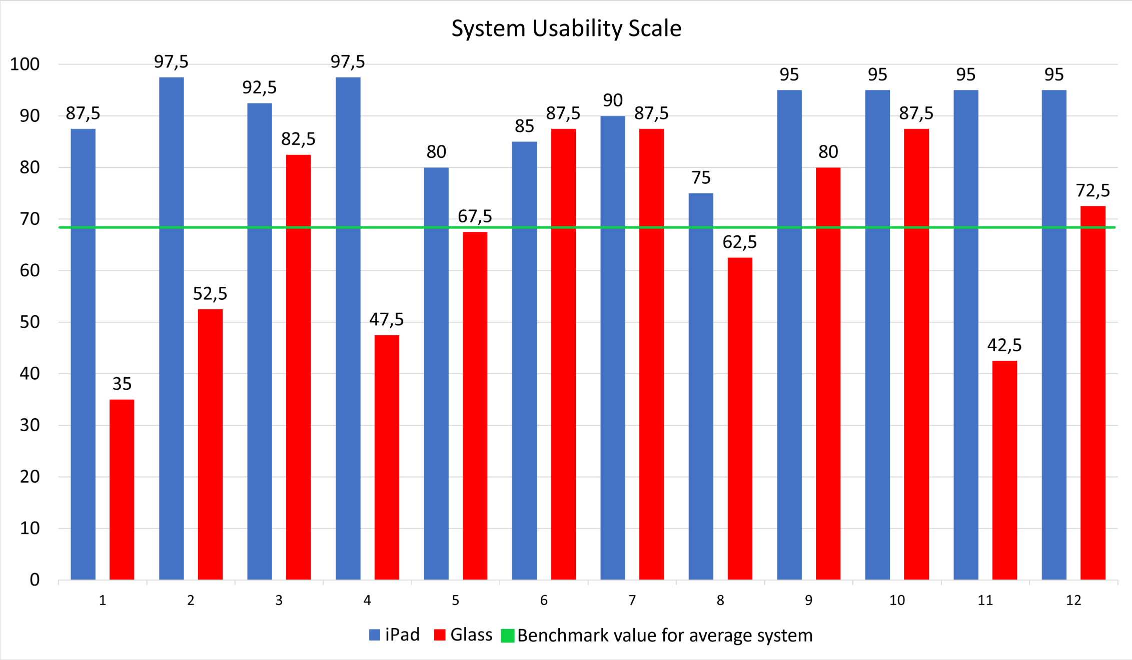


Figure 5.11: Bar diagram of System Usability Scale score from each test subject. Green line represents the 68 mark for an average system

Chapter 6

Discussion

This chapter discusses the results produced by this project. The overall user test and its components are reflected upon. The A/B test and semi-structured interview are thoroughly analyzed and connected to the NASA TLX and SUS evaluations to find answers to why the results ultimately yielded from the experiment.

6.1 A/B Testing

Most tests ran smooth and every subject provided highly informative data that can be used for further application development and implementation. Remarks considering the graphical interface, possible user constraints to prevent errors, subject expectations and feelings towards navigation and controls, and much more was the result from each test session.

The most occurring obstacle the supervisor could observe from the tests was the Google Glass touch pad. Subjects' inexperience with the hardware and intuition of where the touch pad was located, and its dimensions, resulted in unwanted inputs. Users typically had to try several times to get a touch gesture to register and many times the registered gesture was one the user was not aiming for. This only got worse when subjects used two fingers to navigate the application and each subject needed at least three or four tries before a command was registered. Some subjects was perceived to get better at the touch pad further into the test and one user stated "It's all about learning the touch pad, then the rest comes naturally." which could indicate that with time, a user could grasp the controls and become effective at navigating the application and Glass. An early test subject expressed concern that hand moisture could stick to the surface and form a layer on top of the touch pad which prevented gestures to be picked up properly. The glasses were wiped off onwards using a micro fiber cloth before and after each test. This however had little to no effect on the rest of the tests. If the subjects had gotten more time to get used to the Glass touch pad then maybe the experience might have been more positive but this would require either having a longer and more focused on-boarding session, or having each subject borrowing a pair of Glass to test at home. Both of these options seems unreasonable.

Another undeniable problem was the discovery mode and orientation of Glass. With only 50% of the test users being able to use the discovery mode as intended, the current state can only be seen as a failure. With this current implementation it is not a viable way to discover smart devices and should be improved. Problems surrounding the *sensorManager* were observed during the development but thought to have been solved. This was quickly proven to not be the case after multiple subjects were unable to use the sensors to discover any devices. The main issue regarding this was how the orientation was computed. The azimuth of Glass was provided and later compared to earths magnetic field to compute the "true north". Disabling this step of the process gave more stable azimuth values which was not as affected by pitch, roll and yaw as normally. This azimuth did not properly reflect reality, but was consistent enough to work as a way to discover devices for this study. When regarding the 50% who successfully used the feature to discover devices, they were perceived to accept and appreciate the feature and that overall positive experiences were noted. This could indicate the future potential of this method if it could be improved.

Subjects who more or less frequently wear glasses and need to, due to poor eye sight, faced troubles when trying Google Glass. They had to make a decision to remove their own glasses and run the risk of compromising their eye sight, or to wear Glass on top of their regular glasses. This resulted in users either not seeing the Glass display or distant objects properly, or having to awkwardly fit two pairs of glasses on their head. Both options were negative solutions and not how Glass is intended to be used. This is a big negative aspect that probably affected the end result. If subjects applicable for the test was restricted to none glass wearers only, it would probably yield better results. This would however not be fair to the study since poor eye sight occurs among humans in various degrees and cannot be disregarded.

Another physical problem regarding Glass was their overall design and weight distribution. Many of the test subjects had to hold on to the left spectacle frame in order for them to stay put and in a non-crooked angle. One subject reported that the glasses were too big to fit their head, and needed to constantly keep them supported with one hand. Another subject stated that swiping upwards on the touch pad sometimes pulled up that side of the spectacle frame, putting the glasses in an awkward position. Several users noticed the right frame being much heavier than the left and expressing negative feelings towards the uneven weight distribution. Subjects with longer hair pointed out that a risk of their hair getting in the way of the touch pad was apparent and that they should have brought a hair band to the test to be able to use the touch pad more effectively.

Eager test subjects who sent multiple requests after another were more likely to run into communication issues. Bombarding Home Assistant with HTTP requests is a quick way to crash the Glass application since no function had been implemented to handle multiple responses not being received on time. This is purely a mistake made by the author when developing the application. Combining this with a varying strength of WiFi connection, more than one user faced longer response times, a slower application and some even crashed the application. These issues was not as apparent for subjects who took their time with their requests and patiently waited for the application to process each of them. This is not something one can expect from every user however and measures towards this should be a priority when improving the application in the future.

The command controls were surprisingly effective and most users figured them out intuitively or after a few tries. The subjects seemed very comfortable using **SWIPE UP** and **SWIPE DOWN** when increasing and decreasing both brightness and volume. Some subjects needed a few tries to understand that these gestures was used to change brightness, but after learning and considering them, most of them accepted them instantly and saw the logic behind them. Some users argued that this might be a choice of preference but after moving on to the speaker controls, they all accepted the controls and regarded them positively. The fact that every single test subject managed to control the speaker correctly raises a few questions. One explanation could be the transfer-of-learning between Philips Hue and Sonos. Since the subjects were tasked with interacting with the lights first, they might apply the same logic behind the light controls to the speaker controls. If the order of tasks were switched, subjects might not have managed as easily. Another explanation could be that the subjects were faced with two tasks when interacting with the speaker, rather than just one. When faced with only changing one variable, they could do this in two ways which might seem equally likely. But when faced with both a volume change and a song change, the subjects might have determined one action being more likely connected to a certain touch command and concluded that the other, less likely option, was achieved with the other touch command.

The two-finger navigation controls were, as stated before, not as accepted by the subjects and was often forgotten or confused with other commands. Choosing two fingers as navigation was a risk the author was aware of and took some time to decide on. These were not considered as main controls however, since the main way of choosing a smart device was through the discovery mode. These controls are instead seen as "experienced user controls" as short cuts to navigate the application faster. The concerns should not be disregarded however, and further development should try to look into an easier way to navigate manually through devices.

When asked to describe the graphical interface, all users were able to correctly interpret the text fields, indicators and visual aids that were presented in each layout. When describing what they could see on the display, one subject raised a concern that not enough contrast was being used. The user pointed out that depending on what environment the Glass prism was in front of, it could have positive or negative effects on the users experience. For instance, if a test subject turned on the light bulb and positioned their head in an angle that put the Glass prism orthogonal to the light source, the light source would make the white background and application interface hard to see since it shines into its position. If the layout color scheme was matched with the environment in a contrasting way, or further measures were taken to improve the contrast of the interface, the information on screen could be made more visable to the user. Misunderstandings regarding the brightness and volume level was raised from several subjects. It was unclear what range both the brightness level and volume level had. Subjects who experimented figured out that the brightness went from 0-255 and among these most stated they would prefer a 0-100 range instead. The speaker was perceived as too loud to experiment with but subjects guessed this ranged from 0-100 but that they were not certain.

This was intended by the author to explore if a unified expectation could be observed and it concluded that the brightness and volume should range from 0-100 instead of 0-255. All subjects interpreted the green bar as a volume indicator when faced with the speaker layout. This was a positive observation since this bar was not supported by an informative text field explaining its purpose. This might fall under the same transfer-of-learning since the brightness bar was described as "Brightness" and that this layout was presented to the subject before the speaker layout. The author suggests that an informative text field is added to the volume bar to keep the two layouts as similar as possible.

6.2 Semi-structured interviews

The semi-structured interviews proved to be very useful to explore each subjects personal opinions regarding the tests, application and hardware. When asked about how each test went, most subjects had a positive feeling about their performance and most commonly stated the iPad solution as the device and application they performed best with. The explanation given for this was mainly prior knowledge, experience and design. Since no user had any proper experience with Glass, it was directly put in a disadvantage position compared to the iPad which all subjects had prior experience of. The Home Assistant application for iOS looked and functioned similar to other iOS applications, whereas the Glass application was less developed and less familiar. A few subjects stated that Glass could potentially become the better performing tool for the task given, but that more experience was needed in order to feel as comfortable controlling Glass as they felt using the iPad.

When discussing their opinions about Glass and the application, many were positive towards the concept but raised a fair amount of criticism towards Glass' design. This became the common trend between the subjects and something that became impossible to ignore. The design of Glass definitely influenced the overall results due to the perceived awkwardness and bulky design and the focus was often put on the difficulties regarding the touch pad. Issues regarding size, weight, shape and look were common and had negative impact on their performance. This, surprisingly, did not bleed into most subjects perception of the application, which was generally accepted and liked. Constructive criticism were raised and the most occurring ones were to add more contrast between background and informative indicators, better precision when increasing and decreasing brightness and not just stepwise, the classic media player icons pause/play/next/previous on the speaker layout and a better discovery mode.

The discovery mode, despite its flaws, was also met by positive response. The users who could not use it successfully still saw positive potential in the feature. When asked about it, the majority responded that in an unfamiliar environment or setup, the feature would be fun and valuable to use to see what was available to interact with.

Many saw future potential and expressed how they would like to use this at work. One subject stated "If I were to use smart glasses and this type of wireless control at work it would ease my tasks and free up my hands which I use a great deal.". All except one test subject could see smart glasses becoming more popular and widely used in the future and were positive towards using smart glasses themselves if the design improved both in usage but also in aesthetics.

6.3 NASA Task Load Index

When analyzing the result given by the NASA TLX it became clear that Glass was seen as the more difficult device to complete the task with. All but one subject regarded the iPad scenario as the lighter task load with the biggest difference measuring 31.67 points, which is more than double perceived task load between the iPad and Glass for the specific user. Three out of the twelve users perceived the task load using Glass to be more than twice as high as for the iPad. Two out of twelve users perceived the task load being between 64-85% greater for Glass than iPad. The rest of the subjects had fairly equal perception of task load between the two control devices.

A very important thing to notice is that one test subject's answers were not saved properly and is therefore missing. This was due to a technical issue where the answers were not saved after the survey was filled. Given the subjects opinion and SUS evaluation, this could have had a big impact on the overall average score. This subject is not included in the overall average TLX score.

6.4 System Usability Scale

The SUS evaluation further show the iPad as the more favourable device to control smart devices with. Only one out of the twelve test subjects gave Glass a higher score than the iPad. The biggest measuring difference was 52.5 points which was expressed by two subjects. Two other subjects stated a difference of 45 and 50. The rest were somewhat similar with some differences greater than others.

With the average score of 68 indicating an average usability, scores above 68 indicate above average usability and below indicate below average usability. All twelve subjects scored the iPad and its application above 68 making it an above average system according to this test. The lowest score given was 75 and highest at 95. Eight out of twelve subjects rated it 90 or higher. This matched the opinions expressed in the interviews.

The Glass usability score varied heavily between users. Six out of twelve subjects gave scores below 68, some as low as 35. Meanwhile, the remaining six subjects that scored above 68 never scored 90 or higher with highest rating on 87.5.

The lower scores for Glass in the SUS evaluation can most likely be traced back to the negative opinions raised about Glass in the interviews. Subjects issues with the touch pad, awkward design as well as application crashes and unsuccessful use of the discovery mode can all be factors to the lower scores in the SUS evaluation. The above average scores were as many as the lower, but the subjects who did score below average gave very low scores in comparison to how high the above average scores were. This illustrate that the subjects who thought Glass were usable thought they were good, whereas the subjects who thought Glass were unusable, thought they were very bad.

6.5 Future work

For future projects or development it would be interesting to evaluate a different pair of smart glasses. The overall opinion to Glass, as stated by the test subjects, were that the touch pad was difficult to master. Trying a different pair might result in a better user experience than Glass.

Another issue to definitely handle is the discovery mode. In this project, the discovery mode worked in the test environment but was not flexible to work in a natural setting. Trying to explore other methods on discovering smart units, such as computer vision and machine learning approaches, would be very interesting not only from a functional standpoint, but also from a CPU and hardware performance point of view.

Lastly, more smart devices should be implemented and tested. Security cameras and security systems could be the next step in a home environment and seems like an interesting path to explore. If users could get a live feed from their network cameras or unlock their front door using smart glasses, it could probably benefit many private users in their daily life.

Chapter 7
Conclusion

The purpose of this master thesis was to explore and evaluate an alternative technology to wirelessly control smart devices. Goals and research questions were established for the project to answer. An application was developed for Google Glass that let users discover and interact wirelessly with two available smart devices: Philips Hue and Sonos Play:1. The application established a connection between the user and the smart devices, made the user able to send and receive information to and from the smart devices, presented the user with an informative graphical interface and gave the user a way to discover smart devices in an environment. Glass and the application was tested against an iPad and evaluated by twelve test subjects from a usability and work load perspective. Semi-structured interviews were held to dive deeper into each subjects opinions, thoughts and standpoint towards the two test devices.

The results conclude that Google Glass does not function very well as a smart device controller from a hardware standpoint. Users pointed out an awkward touch pad and problematic overall design. However, positive opinions were observed that show future potential and users appreciated the technology and could see a future where smart glasses become a popular tool. The application developed felt intuitive to the test users, fulfilled its purpose and was appreciated. It was however limited to its Google Glass platform and was not seen as useful and easy to use as the tablet application. Finally, a discovery mode was developed for the application that showcase one way to let users discover interactable smart devices in an environment which, when working properly, showed positive potential and was overall likable.

In the end both the NASA TLX and System usability scale indicated that the iPad and its application was the preferred one to use when interacting wirelessly with smart devices. The same trend was observed in the interviews. Subjects who did not like Glass heavily favored the iPad, whereas subjects who did like Glass, did not particularly see its value today and instead speculated that the technology would be of better design and more mature in the future. With this, the research questions have been answered.

References

[1] Sawood Alam, Charles L. Cartledge, and Michael L. Nelson. "Support for Various HTTP Methods on the Web". In: *CoRR* abs/1405.2330 (2014). URL: http://arxiv.org/abs/1405.2330.

[2] Martin Armstrong. "The market for smart home devices is expected to boom over the next 5 years". In: (2022). URL: https://www.weforum.org/agenda/2022/04/homes-smart-tech-market.

[3] Alexander Arvebratt and Fredrik Magnusson. "Using AR gaze-based interaction to control IoT devices". In: (2021). URL: https://lup.lub.lu.se/student-papers/search/publication/9058576.

[4] Ronald T. Azuma. "A Survey of Augmented Reality". In: (1997). URL: https://www.cs.unc.edu/~azuma/ARpresence.pdf.

[5] Tim Berners-Lee, Larry M Masinter, and Roy T. Fielding. "Uniform Resource Identifiers (URI): Generic Syntax". In: *Request for Comments* 2396 (1998). URL: https://www.rfc-editor.org/info/rfc2396.

[6] Edvin Boyner and Jesper Brodin. "Smart Homes: Design & Development of an Application From a User Centered Perspective". In: (2021). URL: https://lup.lub.lu.se/student-papers/search/publication/9053306.

[7] John Brooke. "SUS: A Retrospective". In: *Journal of Usability Studies* 8 (2013). URL: https://uxpajournal.org/wp-content/uploads/sites/7/pdf/JUS_Brooke_February_2013.pdf.

[8] Stephen Carter et al. *HTTP Extensions for Distributed Authoring – WEBDAV*. RFC 2518. Feb. 1999. DOI: 10.17487/RFC2518. URL: https://www.rfc-editor.org/info/rfc2518.

[9] Liam Alexander Colman. *What is Home Assistant?* 2022. URL: https://home-assistant-guide.com/guide/what-is-home-assistant-and-what-can-it-do/#what-is-home-assistant. (accessed: 1.5.2023).

[10] Lynne M. Connelly. *Waterfall Methodology: A Complete Guide*. 2022. URL: https://business.adobe.com/blog/basics/waterfall. (accessed: 29.05.2023).

[11] Mauro Cordella et al. “Durability of smartphones: A technical analysis of reliability and repairability aspects”. In: *Journal of Cleaner Production* 286 (2021). URL: https://www.sciencedirect.com/science/article/pii/S0959652620354342.

[12] Emily Curryer. “IoT in 2023 and beyond”. In: (2023). URL: https://techinformed.com/iot-in-2023-and-beyond/.

[13] Yogesh K. Dwivedi et al. “Setting the future of digital and social media marketing research: Perspectives and research propositions”. In: (2021). URL: https://www.sciencedirect.com/science/article/pii/S0268401220308082?via3Dihub.

[14] Roy Thomas Fielding. *REST API Must Be Hypertext Driven*. 2008. URL: https://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven. (accessed: 29.05.2003).

[15] Google. *About Glass Enterprise Edition 2*. 2023. URL: https://support.google.com/glass-enterprise/customer/answer/9220200?hl=en. (accessed: 28.4.2023).

[16] Google. *Glass Enterprise Edition 2*. URL: https://developers.google.com/glass-enterprise. (accessed: 27.5.2023).

[17] Google. *Glass Enterprise Edition Announcement FAQ*. 2023. URL: https://support.google.com/glass-enterprise/customer/answer/13417888.

[18] Google. *Samples and Demos*. 2020. URL: https://developers.google.com/glass/samples. (accessed: 20.05.2023).

[19] Mark Hall and William L Hosch. “Google”. In: (). URL: https://www.britannica.com/topic/Google-Inc. (accessed: 27.05.2023).

[20] Sandra G. Hart. “NASA-TASK LOAD INDEX (NASA-TLX); 20 YEARS LATER”. In: (2006). URL: https://journals.sagepub.com/doi/abs/10.1177/154193120605000909?journalCode=proe.

[21] Adam Hayes. *Smart Home: Definition, How They Work, Pros and Cons*. 2022. URL: https://www.investopedia.com/terms/s/smart-home.asp. (accessed: 05.05.2023).

[22] HomeAssistant. *Installation*. URL: https://www.home-assistant.io/installation. (accessed: 1.5.2023).

[23] HomeAssistant. *Integrations*. URL: https://www.home-assistant.io/integrations/. (accessed: 1.5.2023).

[24] IBM. *HTTP responses*. URL: https://www.ibm.com/docs/en/cics-ts/5.2?topic=protocol-http-responses. (accessed: 17.05.2023).

[25] IBM. *What is a REST API*. URL: https://www.ibm.com/topics/rest-apis. (accessed: 11.05.2023).

[26] IKEA. *Say hej to IKEA Place*. 2017. URL: https://www.ikea.com/au/en/customer-service/mobile-apps/say-hej-to-ikea-place-pub1f8af050/ (visited on 05/08/2023).

[27] Hiren Kanani. *How do AR Glasses work? Everything You Need to Know*. 2022. URL: https://pluto-men.com/how-ar-glasses-work/. (accessed: 27.05.2023).

[28] Katherine Lazarevich. *The Connectivity Problem in Smart Homes and Ways to Solve It*. 2018. URL: https://www.iotforall.com/smart-home-connectivity-solutions.

[29] Li Li and Wu Chou. "Design and Describe REST API without Violating REST: A Petri Net Based Approach". In: *2011 IEEE International Conference on Web Services*. 2011, pp. 508–515. DOI: 10.1109/ICWS.2011.54.

[30] Zhigang Liu et al. "HTTP Compression Techniques". In: *2005 IEEE Wireless Communications and Networking Conference* 4 (2005). URL: https://ieeexplore.ieee.org/abstract/document/1424906.

[31] Eva Mackamul. "Improving the Discoverability of Interactions in Interactive Systems". In: *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (2022). URL: https://doi.org/10.1145/3491101.3503813.

[32] Natalie Marchant. *What is the Internet of things?* 2021. URL: https://www.weforum.org/agenda/2021/03/what-is-the-internet-of-things/.

[33] Michael Meng, Stephanie Steinhardt, and Andreas Schubert. "Application Programming Interface Documentation: What Do Software Developers Want?" In: *Journal of Technical Writing and Communication* 48.3 (2017). URL: https://www.researchgate.net/publication/318733467_Application_Programming_Interface_Documentation_What_Do_Software_Developers_Want.

[34] Jeremy Miles, Paul Gilbert, and Fiona Fylan. *A Handbook of Research Methods for Clinical and Health Psychology*. Oxford University Press, 2005. ISBN: 9780198527565.

[35] Mozilla. *HTTP Messages*. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages. (accessed: 17.05.2023).

[36] Jakob Nielsen. "10 Usability Heuristics for User Interface Design". In: (). URL: https://www.nngroup.com/articles/ten-usability-heuristics/. (accessed: 27.05.2023).

[37] Nintendo. *Pokemon Go*. 2016. URL: https://www.pokemon.com/us/app/pokemon-go/ (visited on 05/08/2023).

[38] Donald A. Norman. *The design of everyday things: Revised and expanded edition*. Basic Books, 2012.

[39] OpenSource. *What is a Raspberry Pi?* URL: https://opensource.com/resources/raspberry-pi. (accessed: 1.5.2023).

[40] Paul B. Paulus and Bernard A. Nijstad. *The Oxford Handbook of Group Creativity and Innovation*. Oxford University Press, 2019. ISBN: 9780190648077.

[41] Philips. *How Philips Hue work*. URL: https://www.philips-hue.com/en-gb/explore-hue/how-it-works. (accessed: 1.5.2023).

[42] Philips. *Startpaket*. 2023. URL: https://www.philips-hue.com/sv-se/p/hue-white-startpaket%E2%80%933-st-e27-smarta-ljuskallor%E2%80%931100%E2%80%93+-smart-button/8719514289239. (accessed: 10.03.2023).

[43] RaspberryPi. *Raspberry Pi 3 Model B+*. 2023. URL: https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus. (accessed: 10.03.2023).

[44] RaspberryPi-Foundation. *What is a Raspberry Pi?* URL: https://www.raspberrypi.org/help/what-20is-a-raspberry-pi. (accessed: 1.5.2023).

[45] Doug Rohrer, Brandon Sholar, and Kelli Taylor. "Tests Enhance the Transfer of Learning". In: (2010). URL: https://www.researchgate.net/publication/40869048_Tests_Enhance_the_Transfer_of_Learning.

[46] Ignatius Moses Setiadi De Rosal et al. "A Comparative Study MD5 and SHA1 Algorithms to Encrypt REST API Authentication on Mobile-based Application". In: *2019 International Conference on Information and Communications Technology (ICOIACT)*. 2019, pp. 206–211. DOI: 10.1109/ICOIACT46704.2019.8938570.

[47] Dennis Samuelsson. "UbiCompass - IoT Interaction with Wearables". In: (2016). URL: https://lup.lub.lu.se/student-papers/search/publication/8887034.

[48] Sonos. *Control API*. URL: https://developer.sonos.com/reference/control-api. (accessed: 28.4.2023).

[49] Sonos. *Introducing Sonos PLAY:1*. 2013. URL: https://www.sonos.com/en/newsroom/introducing-sonos-play-1.

[50] Sonos. *Set up your Sonos Play:1*. 2020. URL: https://support.sonos.com/en/article/set-up-your-sonos-play-1. (accessed: 18.06.2023).

[51] Nick Statt. *Google opens its latest Google Glass AR headset for direct purchase*. 2020. URL: https://www.theverge.com/2020/2/4/21121472/google-glass-2-enterprise-edition-for-sale-directly-online. (accessed: 18.06.2023).

[52] Adobe Communications Team. *Waterfall Methodology: A Complete Guide*. 2022. URL: https://business.adobe.com/blog/basics/waterfall. (accessed: 29.05.2003).

[53] Daniel Tovesson. "Control your home with augmented reality -". In: (2020). URL: https://lup.lub.lu.se/student-papers/search/publication/9008491.

[54] Vuzix. *Telemedicine through Smart Glasses to Benefit Healthcare Pros, Patients – White Paper*. URL: https://www.vuzix.com/blogs/vuzix-blog/telemedicine-through-smart-glasses-to-benefit-healthcare-pros-patients-white-paper. (accessed: 27.05.2023).

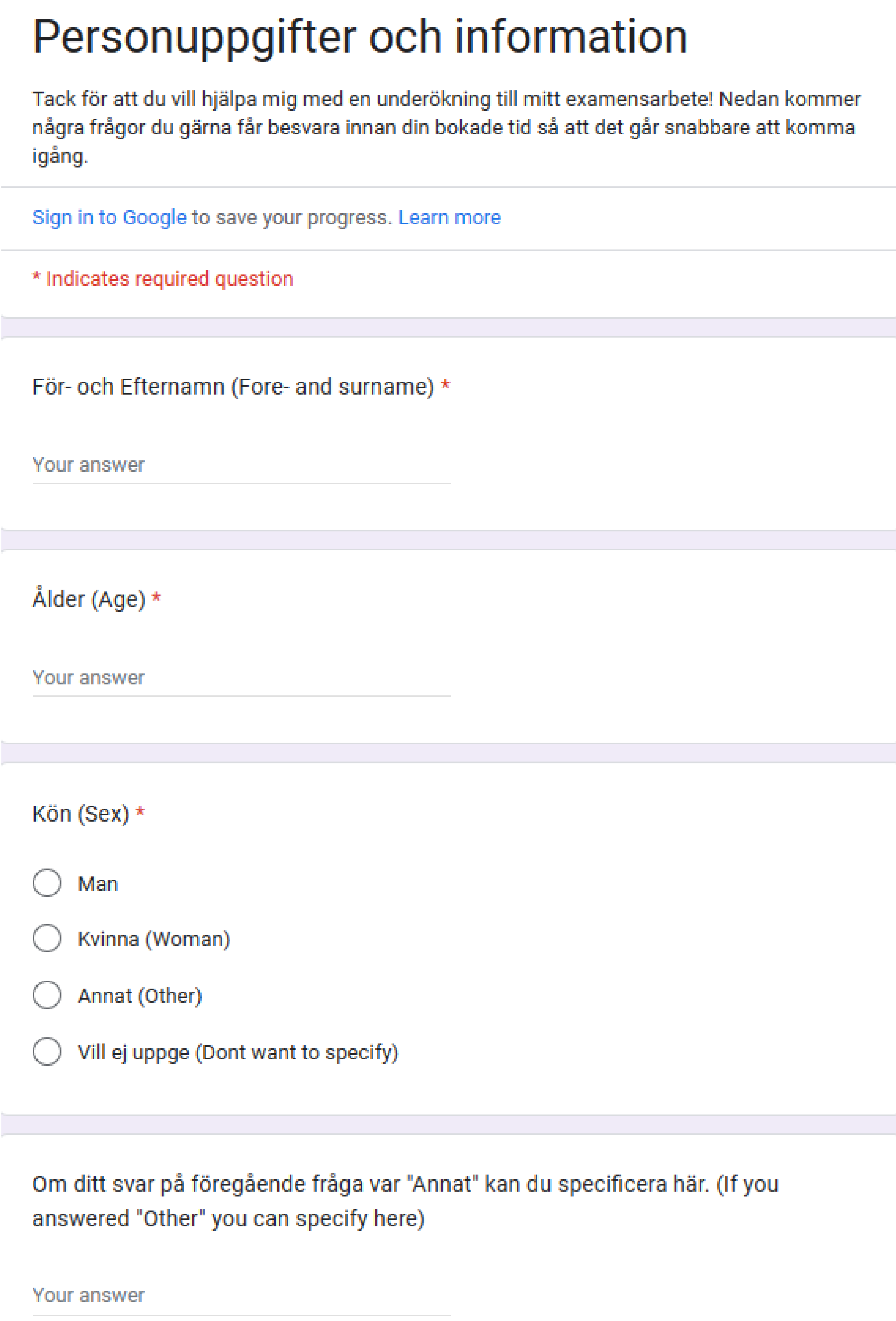
[55] AMA XpertEye. *What is Assisted Reality? Here is what you need to know*. 2022. URL: https://blog.amaxperteye.com/what-is-assisted-reality-here-is-what-you-need-to-know. (accessed: 08.05.2023).

[56] Scott W. H. Young. "Improving Library User Experience with A/B Testing: Principles and Process". In: (2014). URL: https://quod.lib.umich.edu/w/weave/12535642.0001.101?view=text;rgn=main.

[57] Niek Zuidhof et al. "Defining Smart Glasses: A Rapid Review of State-of-the-Art Perspectives and Future Challenges From a Social Sciences' Perspective". In: (2021). URL: https://link.springer.com/article/10.1007/s41133-021-00053-3.

Appendices

Appendix A
Personal questionnaire prior user test



Personuppgifter och information

Tack för att du vill hjälpa mig med en underökning till mitt examensarbete! Nedan kommer några frågor du gärna får besvara innan din bokade tid så att det går snabbare att komma igång.

Sign in to Google to save your progress. Learn more

* Indicates required question

För- och Efternamn (Fore- and surname) *

Your answer

Ålder (Age) *

Your answer

Kön (Sex) *

- Man
- Kvinna (Woman)
- Annat (Other)
- Vill ej uppge (Dont want to specify)

Om ditt svar på föregående fråga var "Annat" kan du specificera här. (If you answered "Other" you can specify here)

Your answer

Figure A.1

E-mail *valfritt* (E-mail *not mandatory)

Your answer

Telefonnummer *valfritt (Phone number *not mandatory)

Your answer

Utbildning: Gymnasium, Högskola etc. Specifiera gärna vilken linje/program/område (Education: High school, University etc. Please specify in what area) *

Your answer

Yrke (Occupation) *

Your answer

Hur teknisk anser du dig själv vara (Skala 1-5). (How techincal would you say you are (Scale 1-5)) *

1	2	3	4	5
◯	◯	◯	◯	◯

Är du bekant med begreppen Augmented Reality (AR) och Virtual Reality (VR)? (Are you familiar with the terms Augmented Reality (AR) and Virtual Reality (VR)?) *

◯ Ja (Yes)

◯ Nej (No)

Figure A.2

Har du någon tidigare erfarenhet med Augmented Reality (AR) eller Virtual Reality (VR)? (Do you have any previous experience with Augmented Reality (AR) and/or Virtual Reality (VR)?) *

- ◯ Ja (Yes)
- ◯ Nej (No)
- ◯ Vet ej (Dont know)

Om föregående svar va "Ja", förklara vad du har använt. (If you answered "Yes" on previous question, please explain what you have used)

Your answer

Har du tidigare erfarenhet av IoT (Internet of Things) eller smarta hemprodukter? T.ex Ikea trådlös, Sonos ljudsystem, Alexa och Google Chrome Cast. (Do you have any previous experience of IoT (Internet of Things) or smart home products such as Ikea wireless, Sonos Sound system, Alexa and Google Chrome Cast?) *

- ◯ Ja (Yes)
- ◯ Nej (No)
- ◯ Vet ej (Dont know)

Har du några IoT eller smarta-hem produkter i ditt nuvarande hushåll? (Do you have any IoT or smart home products in your home?) *

- ◯ Ja (Yes)
- ◯ Nej (No)
- ◯ Vet ej (Dont know)

Figure A.3

Använder du glasögon? (Do you use glasses?) *

- ◯ Ja (Yes)
- ◯ Ibland (Sometimes)
- ◯ Nej (No)

Skulle du vilja ha en kopp kaffe eller te under testets gång? (Would you like a cup of coffee or tea during the test?)

- ◯ Ja kaffe (Yes Coffee)
- ◯ Ja te (Yes Tea)
- ◯ Nej (No)

Vem är din favoritartist? (Who is you favorite musician?)

Your answer

Figure A.4

Appendix B
Test scenario

Test Scenario

You sit down on your couch in your living room with your smartphone/smart glasses connected to your home network. With your smartphone/smart glasses and the installed app you have access to your smart lights and speakers that are connected to your network.

Step 1: Locate the app and give a brief status update on your connected devices.
(What states are they currently in?)

Step 2: You would like to read a book so you decide to turn on the lights.

Step 3: The brightness is a bit too low/high for you to read. Change it to an appropriate setting.

Step 4: After a while you get bored of reading so you turn on some music instead.

Step 5: The volume is a bit low/high. Could you change this to an appropriate level?

Step 6: You feel tired of this song so you want to change the song.

Step 7: Give a brief status update on your connected devices.

Step 8: The dinner was lovely but heavy so you decide to take an early night. Pause the music and turn off the lights.

Figure B.1: Test scenario instructions

Appendix C
NASA TLX questionnaire

NASA Task Load Indicator

Sign in to Google to save your progress. Learn more

* Indicates required question

Name: *

Your answer

Mental Demand: How much mental and perceptual activity was required (eg. thinking, deciding, calculating, remembering, looking, searching)? Scale 1-100 *

Your answer

Physical Demand: How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating)? Scale 1-100

Your answer

Temporal Demand: How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic? Scale 1-100

Your answer

Figure C.1

Performance: How successful do you think you were in accomplishing the goals of the task set by the experimenter? How satisfied were you with your performance in accomplishing these goals? Scale 1-100

Your answer

Effort: How hard did you have to work (mentally and physically) to accomplish your level of performance? Scale 1-100

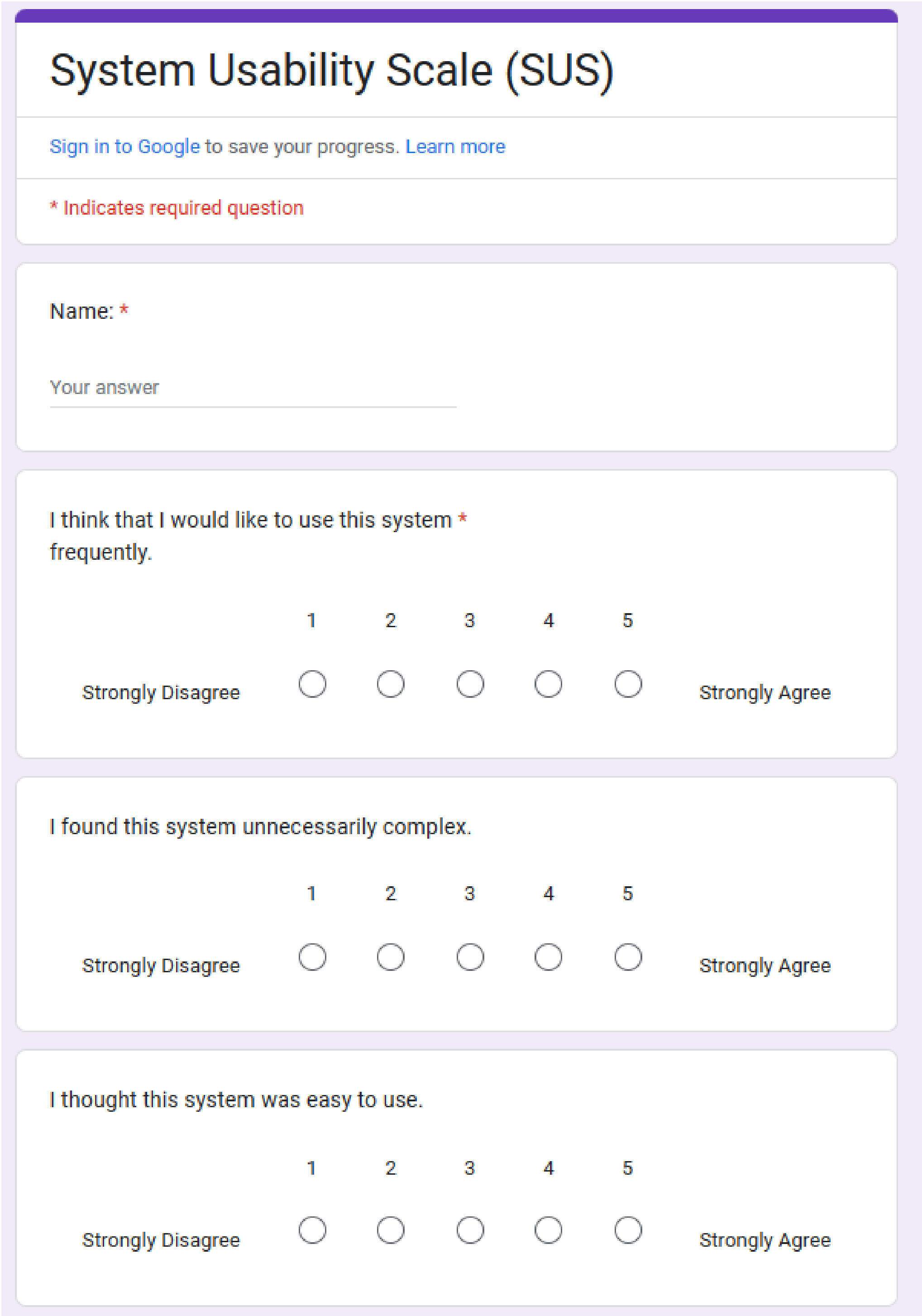
Your answer

Frustration level: How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task? Scale 1-100

Your answer

Figure C.2

Appendix D
System Usability Scale questionnaire



System Usability Scale (SUS)

Sign in to Google to save your progress. Learn more

* Indicates required question

Name: *

Your answer

I think that I would like to use this system *
frequently.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I found this system unnecessarily complex.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I thought this system was easy to use.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Figure D.1

I think that I would need assistance to be able to use this system.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I found the various functions in this system were well integrated.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

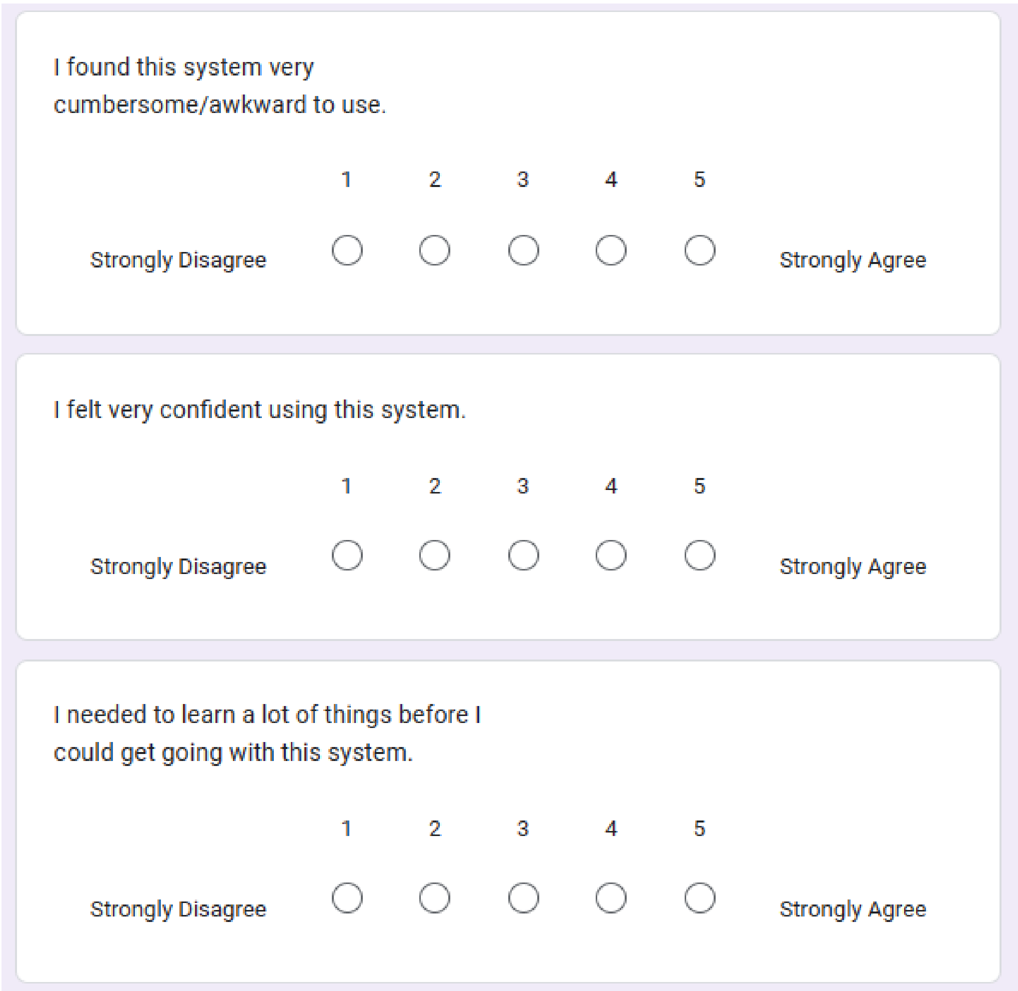
I thought there was too much inconsistency in this system

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I would imagine that most people would learn to use this system very quickly

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Figure D.2


I found this system very cumbersome/awkward to use.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I felt very confident using this system.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I needed to learn a lot of things before I could get going with this system.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Figure D.3

Appendix E
Consent form

Samtycke av datainsamling för examensarbete vid Lunds Tekniska Högskola

Du medverkar i en undersökning för ett examensarbete vid Institutionen för Designvetenskaper på Lunds Tekniska Högskola. Syftet med denna undersökning är att samla in värdefull data till undersökning av ett koncept och dess framtida potential. Testet kommer att spelas in med kameror och mikrofoner för detaljerad dokumentation och underlätta vid utvärdering. Inspelningar och personuppgifter kommer inte att ses av någon annan än projektledare och handledare (Carl Rutholm och Günter Alce). Ditt medverkande i projektet kommer att anonymiseras. Svar och resultat från detta test kan diskuteras i en avslutande rapport. Skulle du ångra ditt medverkande under eller efter testet så har du rätt att återkalla det utan en förklaring. Vi ser självklart helst att du fullföljer din medverkan då den är värdefull för arbetet men skulle du ångra dig under testets gång avslutas det omedelbart och inspelningar samt enkätsvar förstörs. Skulle du ångra dig i efterhand innan arbetet publicerats eller presenterats så kontaktar du Carl Rutholm: ca2742ru-s@student.lu.se för att få dokumentation och deltagande borttaget.

Villkor

- Du har rätt att avsluta testet under vilket skede som helst utan att uppge förklaring.
- Testet kommer att dokumenteras i form av både ljud och bild samt tid för avklarat testscenario.
- Du har rätt att ångra ditt deltagande i efterhand och få all insamlad data borttagen så länge det inte har publicerats, presenterats eller används i annat akademiskt syfte.
- Slutsatser och resonemang kommer att föras kring testets resultat. Dina personuppgifter kommer ej att publiceras men svar och resultat kan komma att citeras i slutlig rapport.

Jag har tagit del av informationen ovan och godkänner villkoren:

Namn: ______________________________

Signatur: ______________________________

Datum: ______________________________

Handledare för undersökning

Signatur: ______________________________
Carl Rutholm

Figure E.1

Appendix F

Home Assistant iOS

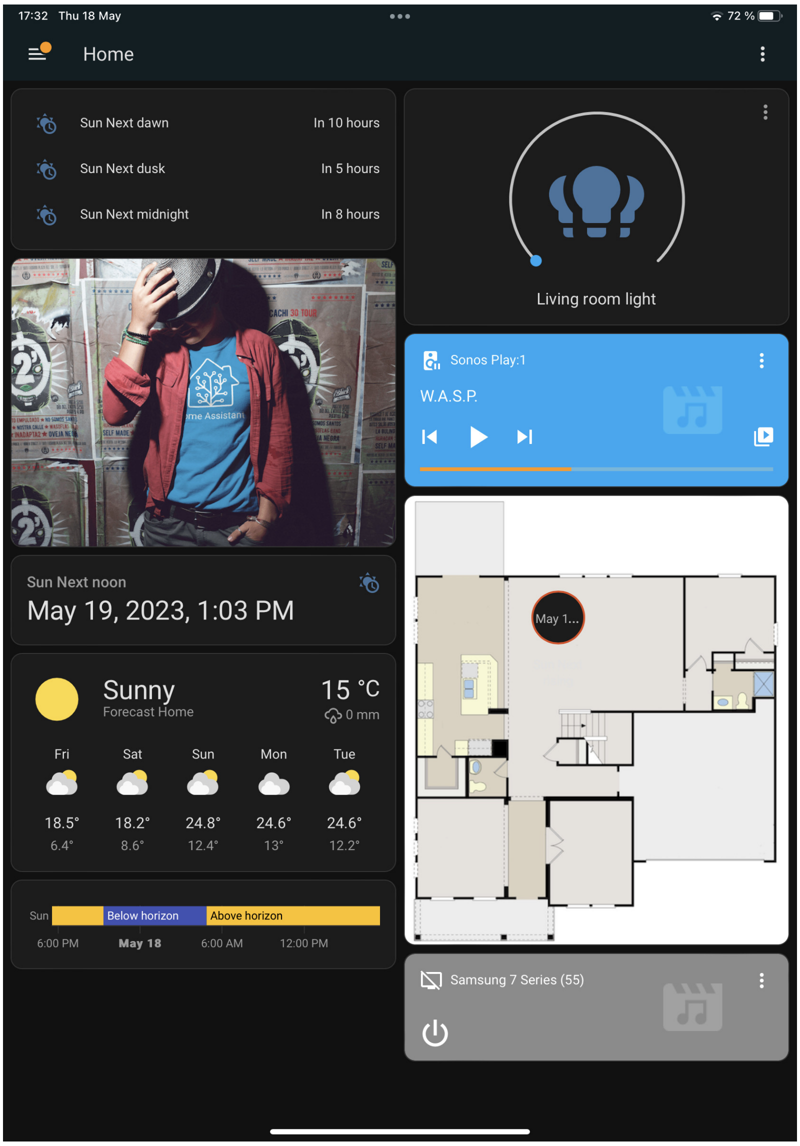

17:32 Thu 18 May
72 %
Home
Sun Next dawn
In 10 hours
Sun Next dusk
In 5 hours
Sun Next midnight
In 8 hours
Living room light
Sonos Play:1
W.A.S.P.
Sun Next noon
May 19, 2023, 1:03 PM
May 1...
Sunny
Forecast Home
15 °C
0 mm
Fri
Sat
Sun
Mon
Tue
18.5°
6.4°
18.2°
8.6°
24.8°
12.4°
24.6°
13°
24.6°
12.2°
Sun
Below horizon
Above horizon
6:00 PM
May 18
6:00 AM
12:00 PM
Samsung 7 Series (55)

Figure F.1