Kunnskap for en bedre verden

Department of Electronic Systems

TFE4930 - Master Thesis

A web-based application for QoE evaluations

Author:
Fredrik Skorstøl

Supervisors:
Andrew Perkis and Asim Hameed

16th June 2022

Abstract

In this project, the aim is to create an application that combines and presents data from evaluation of XR experiences. The objective is to make the process easier and more efficient for the conductor of an extended reality experiment, filling the gap between testing and evaluation.

The use of extended reality is growing rapidly within multiple platforms, applications, and development fields. The technology has proven to be helpful for simplifying and streamlining situations and problems related to the real world, creating innovative and forward-thinking solutions. Different subjective and objective evaluations are being used to ensure that performance quality is as good as possible, which often requires numerous manually conducted steps. The application will include various evaluation methods where multiple of them can be chosen for the same experiment, according to whatever the user needs. Techniques selected are determined according to the most frequently used evaluation procedures of virtual reality experiments and studies. After the toolbox is up and running, it will be put through its own evaluation by being rated by a selection of people using a usability scale. The test results show that the app will need more work in order to enhance performance and user experience. Furthermore, some aspects concerning data protection and security encountered when designing and implementing an application dependent on user information will be addressed.

Sammendrag

Målet for prosjektet er å utvikle en applikasjon som kombinerer og presenterer data fra evaluering av XR opplevelser. Hensikten er å gjøre prosessen lettere og mer effektiv for utføreren av eksperimenter relatert til utvidet virtuelle virkeligheter og dekke til det eksisterende spriket mellom testing og evaluering.

Bruken av utvidet virtuell realitet vokser raskt innen ulike plattformer, applikasjoner, og utviklingsfelt. Teknologien har vist seg å være nokså behjelpelig for å forenkle og effektivisere hendelser og problemer relatert til virkeligheten ved bruken av innovative og fremtidsrettede løsninger. Forskjellige subjektive og objektive evalueringer blir brukt for å forsikre at ytelseskvaliteten er best mulig, noe som ofte krever at en rekke manuelt utførte steg. Applikasjonen vil inkludere forskjellige evalueringsverktøy, der flere av dem kan bli anvendt samtidig for samme eksperiment, avhengig av hva brukeren har behov for. Evalueringsmetodene som er inkludert er valgt med hensyn på de metodene som oftest blir anvendt innen virtuelle realitets eksperimenter og studier. Etter at en prototype av applikasjonen er fungerende, har den blitt testet og vurdert av en gruppe selekterte personer ved hjelp av en brukervennlighets skala. Test resultatene viser at applikasjonen trenger å jobbes mer med for å øke applikasjonsytelse og brukeropplevelsen. Videre vil enkelte aspekter vedrørende databeskyttelse og sikkerhet som oppstår ved utforming og implementering av en applikasjon som er avhengig av brukerinformasjon, bli tatt opp.

Table of Contents

List of Figures

List of Tables

1 Abbreviations

IMEx	Immersive Media Experience	1
VR	Virtual Reality	1
IPQ	Igroup Presence Questionnaire	1
PQ	Witmer Presence Questionnaire	1
ITC-SOPI	ITC sense of presence inventory	1
ITQ	Immersive Tendency Questionnaire	1
XR	Extended Reality	1
MR	Mixed Reality	1
QoE	Quality of Experience	1
AR	Augmented Reality	1
HMD	Head Mounted Display	2
SSQ	Simulator Sickness Questionnaire	5
SUS	System Usability Scale	5
EDA	ElectroDermal Activity	5
MOS	Mean Opinion Score	4
PSNR	peak signal-to-noise ratio	6
SD	Standard Deviation	6
TSV	Tab Separated Values	9
OpenCV	Open Source Computer Vision Library	18
POPD	Protection of Personal Data	23
ZAK	Zoom Access Key	10

POPD	Protection of Personal Data	23
GDPR	General Data Protection Regulation	23

2 Introduction

Today's society is totally dependent on digital media, which has become part of our everyday culture. The constantly increasing influence of digital media is caused by the world's technological advancement, changing how we educate, entertain, and interact with each other. People want more out of their multimedia experiences and demand a higher level of engagement and immersiveness, which eventually have made the concept of Extended Reality (XR) to emerge and get a solid foundation in the media industry. XR is the collective designation used for the different partially or fully digitally generated realities and technologies that exist. This includes Mixed Reality (MR), Augmented Reality (AR), and Virtual Reality (VR), all with the different capabilities of capturing a person's attention toward digital objects or environments. The three different realities can be illustrated as the simplified representation:

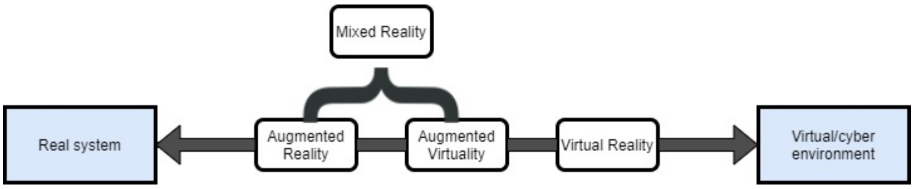


Figure 1: Reality-Virtuality continuum

Source: Milgram, Takemura *et al.* 1995

With the excessive growth of extended reality environments and interactive media, multiple definitions, models, and frameworks of immersion have emerged. The discussion around quality and the factors used to "measure" this concept needs to be extended to avoid misconceptions as it closely relates to the term presence. This topic is further discussed in the white paper by Perkis, Timmerer *et al.* 2020, which provides accurate definitions for immersion and presence, leading to a complete definition of Immersive Media Experience (IMEx). These concepts are the background for how immersive media are evaluated, both for subjective, objective, physical, and psychological measures. The measures are directly related to a users Quality of Experience (QoE) when encountering a system or service Brunnström, Beker *et al.* 2013.

User-perceived QoE can better be understood using the framework presented in Brunnström, Beker *et al.* 2013. It describes experienced quality as a result obtained from comparison and judgment processes, in which perceived quality features are compared to what's expected by the user. This quality expectation, in terms of immersive media, correlates with presence questionnaires(Igroup Presence Questionnaire (IPQ), Witmer Presence Questionnaire (PQ), ITC sense of presence inventory (ITC-SOPI) are some examples) that are well-defined within the field of multimedia assessment. A subjective method such as this may give reasonable indications of how virtual reality can mediate content and increase the feeling of presence. As described by Witmer and Singer 1998, presence in a VE depends on a person's attention when going from the physical world into the VE. This shift does not necessarily mean that their attention requires total displacement from their actual surroundings. Humans will experience different levels of presence in a real setting because of their mentality, affected by memories, daydreams, and thoughts.

The immersiveness of multimedia content will also closely relate to the quality experienced by a user. It is primarily obtained by subjective methods, where questionnaires such as the Immersive Tendency Questionnaire (ITQ) are being used Bektaş, Thrash *et al.* 2021. As immersion is defined by Perkis, Timmerer *et al.* 2020 as the degree to which immersive media environments manage to sub-merge a user's perceptual system in computer-generated stimuli. The more the environment is capable of doing so, the more immersive it is.

The degree of both presence and immersion is often determined by subjective feedback, either verbally or in writing. Using physiological measures may provide additional information on how a system performs in terms of these factors. These measures can, for instance, be a person's heart rate, galvanic skin response, or eye dilation. The overall user experience will, either way, need a combination of methods to be adequately evaluated.

When a system is evaluated during an experiment, different methods and approaches are used to obtain users' impressions and thoughts. The paper presented by Schatz, Sackl *et al.* 2017 is mainly focusing on subjective evaluation and analysis based on results from questionnaires. In De Simone, Li *et al.* 2019 both subjective and objective methods are taken into account, where test subjects are being recorded throughout the entire process. In addition, they conduct a post-experiment interview that is recorded as well.

Other experiments have more focus on objective and physiological evaluation techniques, especially the ones who experiment with novel designs. Tsetserukou, Sato and Tachi 2010, Scheggi, Meli *et al.* 2015, Boletsis and Cedergren 2019 are some examples. Evaluation of studies and experiments involving people's health and how exercising in VR environments can substitute traditional training Kojic, Sirotina *et al.* 2019 will highly benefit from these assessment methods.

The desired goal for this project is to design and create a web-based toolkit used for QoE evaluation, especially targeting experiments conducted with immersive media formats or content. The toolbox will provide multiple methods commonly used for evaluating user experiences, both how content is perceived and its immersiveness. Most evaluation processes today are quite time-consuming because they require a lot of manual work. The application may optimize experiment procedures, provide centralized access to data, and bring efficiency to an otherwise cumbersome activity. The idea is to use the research conducted in Skorstoel 2021 and create a web-based application that provides some of the techniques used, both subjective and objective. Much of the work will be put into merging everything into one common platform to give the user their desired output or service.

The toolkit will provide a collection of methods often used for evaluation purposes. The experimenter will be able to use the application to capture subjects' perceptions and physical signatures for different immersive experiences and have complete control over the data collected, all in one place. The first part of the report will include some related work to get a closer look at the methodology and techniques frequently used, as well as look into some novelties and how they are evaluated. Further, the toolbox will be presented, both its applied methods, design and structure. The application is put through its own evaluation process after a prototype is ready to assess its unfriendliness and appeal. Since evaluation procedures often require humans to participate in experimental activities, it is important to discuss relevant privacy and ethical issues. This topic will be covered in the POPD and ethics chapter with some general guidelines regarding data storage and usage and how the toolbox is designed to prevent such issues. All results will be presented, highlighting the toolbox implementation and UI. The last section discusses the results and problems encountered, together with a conclusion and suggestions for future work.

3 Related work

This chapter is included to give a better comprehension of the term immersive media. It comprises the technology applied in extended reality environments and influencing factors towards immersive media experiences. With current knowledge, a toolkit like this does not exist for user experience evaluations. It usually takes some effort to involve different evaluation methods, so providing the user with a tool to gather all their experiment information and data will be quite beneficial. Since there are few or no such solutions available to compare with, the theoretical research is focused on how evaluation procedures usually are conducted and what kind of data are relevant for measuring immersiveness, performance, and user's overall experiences.

3.1 Immersive media technology

The technological advancement within immersive media has led to an increasing number of XR equipment manufacturers. The products come in different shapes and sizes depending on which reality they are designed for. The most vital device for immersive media presentation is the Head Mounted Display (HMD). The display is placed in front of your eyes, projecting either fully or partially virtual objects and environments.

The main device categories used when immersed in extended reality environments are listed in Table 1, where some of the most popular products are mentioned.

Table 1:
*many of the HDM's can be used for both MR and AR applications

Immersive media devices			
	Mixed reality	**Augmented reality**	**Virtual reality**
Hand-Held devices	Smart phones and tablets	– ‖ –	– ‖ –
HMD devices	Microsoft HoloLens	Microsoft HoloLens	Samsung Odyssey
	Zappar ZappBox	Raptor	Oculus Quest
	Magic Leap One	Magic Leap One	Valve index VR kit
	Nreal Light	Moverio BT-300	HTC Vive Pro
Spatial devices	Lumen	LightForm	Barco

Hand-held devices may also include controllers, motion sensors, and tracking equipment, often used in combination with HDMs or projectors. HDMs used in mixed or augmented reality can be lightweight goggles or see-through displays, allowing the user to see the physical environment while wearing the device. The HDMs designed for VR experiences block out all visual and/or auditory senses, fully capturing the user in the data-generated world.

3.2 IMEx

Over the past decades, immersive media have brought a lot of attention to human-computer interactions, where either the user is immersed inside the virtual or augmented space, or computer-generated artifacts are projected onto real-world objects/surfaces. Creating wholly or partially digital environments invokes a user's feeling of being present, which is defined by Witmer and Singer 1998 as *"The subjective experience of being in one place or environment, even when one is physically situated in another"*. To fully comprehend an immersive media experience where the concepts of immersion, presence, and immediacy are fundamental, the following features are used to characterize immersive media(Perkis, Timmerer *et al.* 2020):

- *Immersivity*: Combination of sensory signals essential for user space and engagement.
- *Interactivity*: User interactions with computer generated effects and avatars through an interface.
- *Explorability*: Users opportunity to move freely around and discover the virtual/augmented world.
- *Believability*: Fidelity and validity of sensory features within the generated environment.
- *Plausibility*: Coherence and consistency of symbolic features users perceive within the generated environment.

With these features in mind, immersive media can be defined as:

"a high-fidelity simulation provided and communicated to the user through multiple sensory and semiotic modalities. Users are emplaced in a technology-driven environment with the possibility to actively partake and participate in the information and experiences dispensed by the generated world."

When establishing the characterization of IMEx, it is also important to have a definition for QoE of immersive media. A working definition of QoE is developed in Brunnström, Beker *et al.* 2013 as:

"QoE is the degree of delight or annoyance of the user of an application or service. It results from the fulfilment of his or hers expectations with the respect of the utility and / or the enjoyment of the application or service in the light of the user's personality and current state.

In this context, The current state may refer to situational and temporal changes in a person's feelings, way of thinking, or behavior. These changes may be caused by specific events experienced or the content of the experiment itself.

A relationship between IMEx and QoE for immersive media can be established as it is commonly assumed that feeling presence is essential for QoE of immersive media. The concept of feeling self-location in a virtual environment is often subsidized with the sense of virtual embodiment, including users taking ownership of their own virtual body and feeling in control of their actions when moving and interacting with the surroundings. How the interaction between user and virtual objects is experienced will significantly impact the overall QoE. Immediate and realistic interactions can lead to a high-quality experience, while unrealistic events can cause discomfort and distress Perkis, Timmerer *et al.* 2020.

3.3 Evaluation

Most studies and experiments rely on some sort of reported feedback from the test subjects to determine their QoE and use additional objective measures to get a fuller and more complete evaluation. The same goes for assessing prototypes and novelties where immersive traits are to be captured.

The most common approach when considering subjective measures is through rating scales and Mean Opinion Score (MOS). This type of evaluation is almost always used for experiment assessment since questions can be fitted to the exact purpose of their research goal. Take, for instance, the research performed by Schatz, Sackl *et al.* 2017; the experiment investigates the stalling effects of omnidirectional video streaming and compares it with the same effect added to traditional 2-d video streaming. For assessment, they rely on participants' feedback in questionnaires and use the results to decide the impact it has on the users' QoE. The questions they use are adopted from the IPQ and are used to further analyze their main research questions.

Another example where questionnaires provide the wanted results for evaluation is the research done by Kojic, Sirotina *et al.* 2019. They are exploring the influence of user interface complexity experienced during VR exergames and use both existing and self-defined questionnaires for this purpose. Interviews are less common than surveys, but the method is still used when the proper equipment (voice recorder, camera, ...) and sufficient time and resources are provided. The interview method is used in De Simone, Li *et al.* 2019 together with questionnaire scales to get more in-depth feedback.

In Keighrey, Flynn *et al.* 2017 the results of an experimental study compare users' QoE of an interactive and immersive speech and language assessment implemented in both AR and VR. The user QoE is captured by using both subjective and objective methods. Objective data are collected using equipment for measuring heart rate and EDA, and subjective data involves the users' self-reported ratings in questionnaires both post-test and in-game. The test established by Anwar, Wang *et al.* 2020 does not consider objective measures but instead focuses on user ratings from questionnaires collected during testing. They investigate the effects of stalling in 360-videos under different bit-rate levels and use the ratings to determine the severity of different stalling event scenarios.

Proper evaluation is necessary when testing or developing novel products or prototypes to decide if they are competitive enough to substitute or co-exist with traditional products and equipment. Katsigiannis, Willis and Ramzan 2018 proposes a design and evaluation of a smart-exercise-bike VR system that immerse the user in a virtual environment using commercially available HMD. The solution is proposed as an alternative to regular handheld controllers and steering wheel modules. Both subjective and objective measures are taken to see the effects and correlation between different rendering qualities and simulator sickness the system may cause. The study results showed that users reported a higher feeling of simulator sickness than others who have conducted similar experiments, which is most likely caused by a higher level of sensory conflicts. They exploit both user reported ratings obtained from the Simulator Sickness Questionnaire (SSQ) and take measurements of their galvanic skin response during the entire procedure for assessing this phenomenon.

The paper by Bektaş, Thrash *et al.* 2021 presents a chair-like embodied interface composed of two separate parts, allowing the user to move their legs independently while seated. Like most other motion and navigation prototype solution studies, it investigates the degree of simulator sickness the system may cause compared to others within the same category. To test its usability, they compare it with a game-pad in terms of performance measures like accuracy and time, as well as the subject's body sway and head movement. Most of their testing involves reported feedback acquired by questionnaires. They focus on determining and measuring presence, usability, workload, and simulator sickness by combining performance measures, body posture, and results from questionnaires.

4 Methodology

There is a high diversity of methods used for evaluation procedures because of the variations in how experiments are conducted and in the type of technology or service explored. As mentioned in Section 2, the testing and evaluation processes are performed in multiple ways. Some highly depend on objective metrics and use measurable data to calculate and analyze their results. Others prefer subjective approaches, where they may obtain a more quantitative representation of users' QoE.

This chapter elaborates on numerous evaluation methods to understand the different procedures and equipment used for evaluating immersive media content. A selection of them is thoroughly analyzed, as they will be implemented in the toolbox application. The prototype phase can begin once the selection has been made. A detailed description of design and data formats are laid out as a motivation for choosing the most suitable SW and programming languages to use for toolbox implementation. The final application solution is presented concerning structure, database storage, UI/design, and integrated evaluation tools, including the problems encountered along the way. The section rounds up with some collected user feedback regarding toolbox UI using the System Usability Scale (SUS).

4.1 Survey

Most of the literature used for surveying the evaluation methodologies used for immersive media has been mentioned in Section 3. The ones that are more frequently used between studies and experiments are considered more informative for this project, so they are the ones chosen for further investigation.

From a physiological perspective, objective data may be the participant's heart rate, ElectroDermal Activity (EDA), eye dilation and movement, muscle activity, and other physical outputs that can provide useful information regarding the participant's QoE. Such approaches are often preferred because the objective data are easily measurable and can be presented as an average or mean score. The downside of this method is that the evaluation results may be affected by the subject's stress level, comfort during testing, state of mind, and previous experience with the technology used. These factors should be considered when working with physiological measures, at least to a certain degree.

Apart from the physiological, other known objective approaches are mathematical models and algorithms based on how humans visually perceive media content. The method does not need physical human involvement, removing uncertainties about people's mindsets and feelings. A commonly used quality assessment metric is the peak signal-to-noise ratio (PSNR), which makes a pixel-by-pixel comparison between media content and some reference structure used in the experiment.

Analytical data are also regularly used to capture users' QoE objectively. Time, attempts, or repetitive actions(mouse clicks, body movement, etc.) used for solving separate tasks within an experiment, can often indicate satisfactory or poor performance of a system. This method is often deployed when evaluating video games, where user experiences can be captured by analyzing how frequent the game is used, how many are playing it, the scores users get, the number of stages they manage to pass, or the difficulty of each stage in terms of attempts and deaths. The analytical measures may indicate the game's level of immersion as well as how the overall experience is received.

Available subjective metrics depend on user feedback, often captured through surveys. Such surveys may include several questionnaires designed to answer different topics(immersion, presence, simulator sickness, enjoyment, usability, and so on). They are carried out either before, during, or after the duration of an experiment, depending on their purpose and content. The collected data are often presented in terms of MOS and Standard Deviation (SD) to reflect the quality of the experiment.

A subjective approach can also have some unfavorable effects on the evaluation. During-test questionnaires require pausing the experiment, interrupting the participant, and affecting their experience. Results from a post-test questionnaire can also affect the quality assessment, as users may partially forget how they perceived or felt about certain media content, leading to inconclusive test results. It is also worth mentioning that using MOS to present the feedback removes the diversity from the subjective rating and will only give a collective insight into the user's perspective.

Observational data such as a person's movements and body postures is especially helpful during experiments that include new devices and novel designs. Recording videos or taking images are typically used to capture people's reactions and motions during an experiment session. Video recordings are practical for longer sessions as they may be used to analyze changes in a person's body language throughout the test. Decreasing enthusiasm revealed from bodily radiance may indicate that the experience is getting boring or tiresome. At the same time, a high level of engagement may imply that the person is having fun and that the system brings a high QoE.

The methodologies and techniques investigated in Skorstoel 2021 covers a variety of what people are using for evaluation:

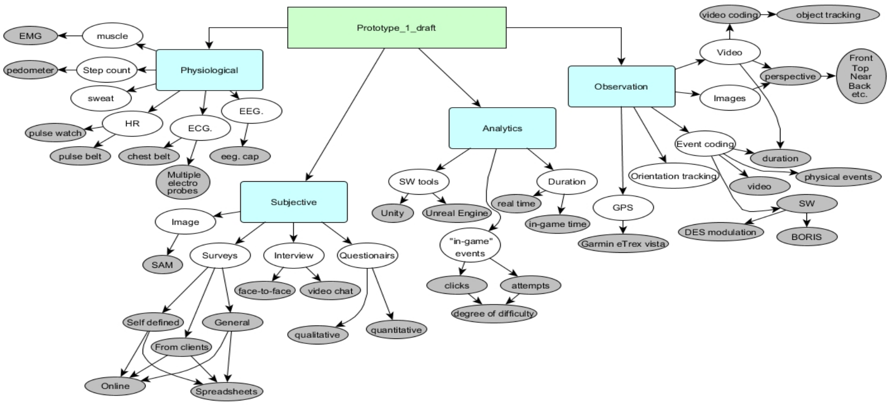


Figure 2: Evaluation methods for immersive media

Source: Skorstoel 2021, p.3

Due to time limitations, only a selection of them is considered for the prototype toolkit. The ones that are intended to use are the most commonly applied among immersive media-related research and experimentation.

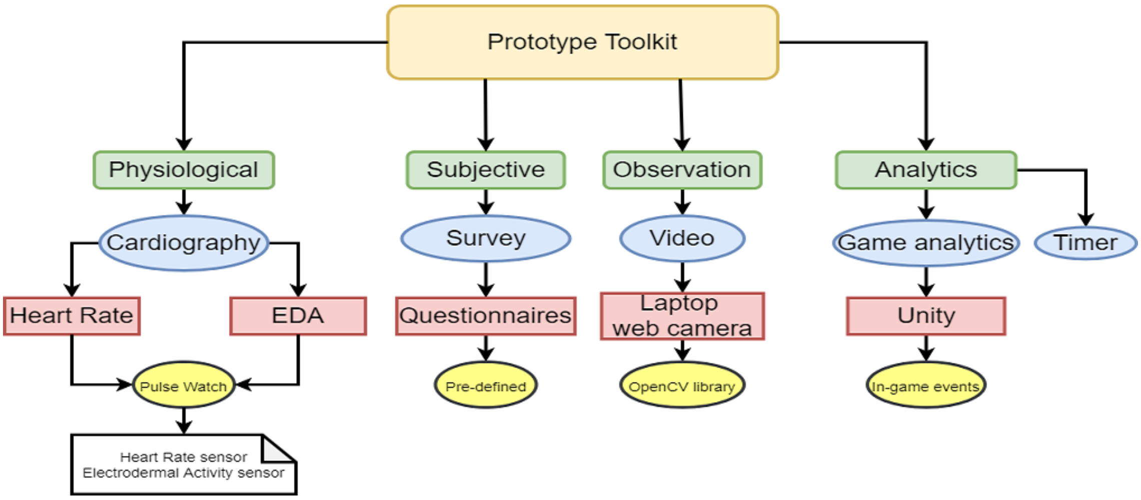


Figure 3: Toolkit evaluation methods

The data construction and formats included in these procedures are considered when designing the toolbox, both regarding what kind of equipment to use and what type of software and programming tools to apply. The application framework will process the data inputs to manage conflicting types and structures so that storage and transmissions can be appropriately handled. To make the toolbox easy to use and provide a user-friendly UI, the design needs to be simple so that the data output can be collected and displayed effortlessly.

4.2 Prototyping

The prototyping process is essential for the development of a product or service. The first phase towards realizing the toolbox application is sketching or digitizing the idea and creating a block diagram of the initial structure. The next step is to determine what's expected as inputs and outputs from the selected evaluation tools to decide which programming language and software framework to use for implementation. The implementation itself can begin once this is resolved. Security risks, ethical concerns, and toolbox user flow are kept in mind while creating the application.

4.2.1 Design

The idea is to develop a minimalistic design so that the user easily and quickly can orient themselves within the application. The first draft is created on paper and later digitized in a prototyping tool. The initial block diagram is given in the figure below.

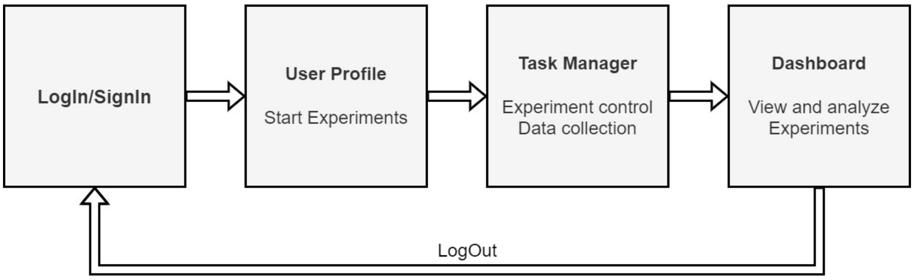


Figure 4: Draft of prototype design

The toolbox should to have:

- An **Index page**, which includes sign up and login logic
- A **User Profile** so people can navigate through the app and start experiments.
- A **Task Manager** for controlling, surveying, and collecting experiment data
- A **User Dashboard** so people can go through all experiments and analyze them.

To get a better sense of the toolbox's appearance and logic, it is realized using Figma (using the block diagram as reference).

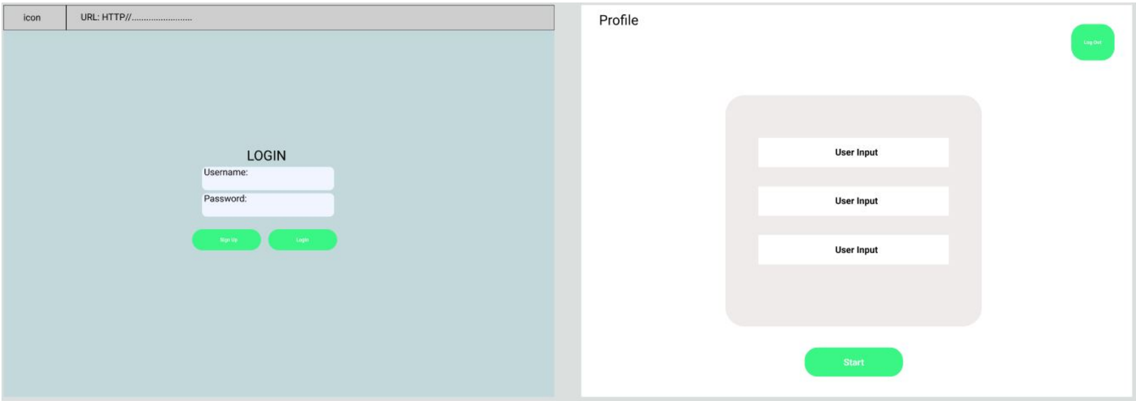


Figure 5: Prototype draft in Figma (Login and Profile)

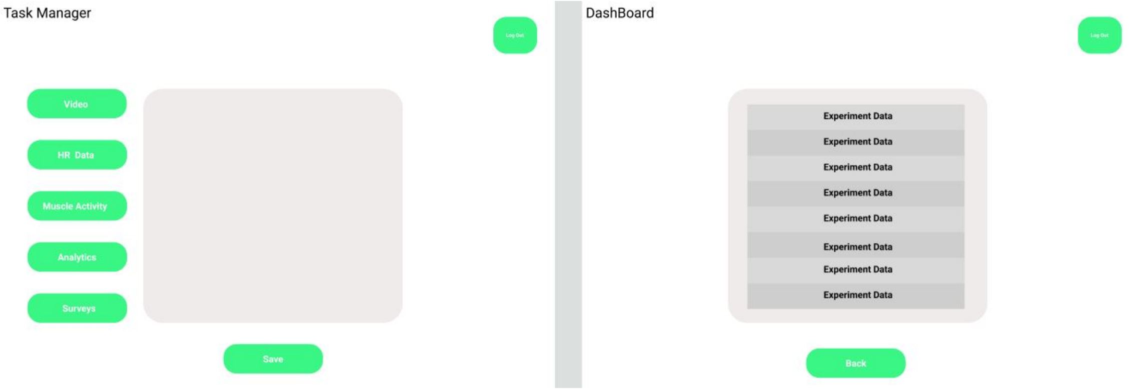


Figure 6: Prototype draft in Figma (Task Manager and Dashboard)

The drafted design is discussed with the supervisor to ensure all the logical aspects are covered and the design itself is suitable. The project supervisor purposed some changes which resulted in the final design presented in figures 11 to 22

4.2.2 Data I/O

From the prestudy in Skorstoel 2021, the following equipment and SW tools where explored in terms of in- and output formats

Table 2: Data formats
*require ip_webcam app(e.g. on smartphone)

Data formats			
Evaluation	**Equipment/SW**	**Input**	**Output**
Heart rate:	**Apple watch**	XML-file	[Time stamp, HR_data]
	Fitbit	[user_id, date, period]	[Time_stamp, HR_data]
ECG:	**Apple watch**	XML-file	[Time_Stamp, $\frac{HR}{time}$, Average]
	Fitbit	[user_id, date, period]	[Time_Stamp, $\frac{HR}{time}$, Average]
Video recording:	**Zoom**	desired settings	link to recording
	OpenCV	[video type, resolution, fps, dimension]	mp4/avi file
	IP webcam*	[resolution, focus, colour effect, fps limit, mode, ...++]	mp4 file
Analytics:	**UE**	JSON/CSV	[nr. of keyboard clicks, event timer]
	Unity	JSON/ Tab Separated Values (TSV)	[nr. of keyboard clicks, event timer]
Muscle measurements:	**EMG**	CSV	[Voltage diff., time]
Survey:	**Questionnaires**	Table(TSV,CSV)	CSV[question,answer]

Cardiography

Both Fitbit and Apple watch are popular and well-known devices that feature heart rate and ECG measurements. None of them are open-source products, but Fitbit provides a complete web API that can be used to retrieve data collected by the sensors integrated into their watches. It is available to use for own application development, as long as it complies with Fitbit's terms of service and data policy and that users consent to share their data(Fitbit 2022). Apple is also providing developers with a framework interface called the HealthKit API. The API can be used for communication and sharing health data between third-party applications and Apple Health(Thryve 2021). Apple Health stores all health-related data captured by an Apple watch locally on the user's device, meaning the data are unavailable from any cloud service.

Evaluations regarding muscle activity are not further analyzed, primarily because of lacking equipment but also because of little knowledge of exactly how muscle data are collected during an experiment session. The evaluation method will not be included in the toolbox.

Video

There are several available devices and software tools used for recording videos. In the pre-study

Skorstoel 2021 only a selected few were surveyed.

Zoom where considered as it offers an SDK designed to let developers and customers build apps using Zoom's communication solutions or integrate the technology into their own created products/services. To enable SDK development, you are required to have a zoom account and an access key(Zoom Access Key (ZAK)) used for app identification and authentication(Zoom-Video-Communications 2022). The ZAK is obtained by request from the zoom admin user and will enable developer privileges for your zoom account.

The OpenCV library includes a common infrastructure for computer vision applications integrable with multiple programming languages and supports different operating systems. The library is a powerful tool in terms of machine learning algorithms but can also be used as a simple video recording service.

By using an IP web camera mobile application, you can record videos using the camera on your smartphone. The video stream can be connected and accessed by the OpenCV library without any restrictions in terms of authentication or validations. You will, on the other hand, need the URL the IP camera is streaming to.

Analytics

Unity and Unreal Engine are two solutions explored for providing an analytical input to the toolbox. Unity features an end-to-end analytics SDK solution for data analysis which can be accessed through the Unity development platform. It is also possible to connect your Unity project to a web-based service using database structures and URL post- and get requests. Unreal Engine does not provide its own analytical service, so a third-party software dedicated to this purpose needs to be included(Apsalar, Flurry,...)(Epic-Games 2022). The game engine is integrated with an analytics blueprint plugin that can be enabled to translate API calls and transfers them to the analytics provider/service you have chosen for the project.

Forms

There exist lots of SW tools for creating questionnaires and scales. The ones considered for this project are Microsoft Forms and Google Forms. Both are universally known services and let customers create their web-based surveys with different specs and appearances. Web forms are preferred over hand-written or similar solutions because they can easily be included and shared within the toolbox application.

4.2.3 SW and Programming language

Language and framework

There are several choices to be made before starting the actual implementation. Python is selected as the primary programming language as it offers a broad spectrum of features that can be utilized for web application development. It is also quite complex and advanced in terms of high-quality and functional programming, making it more suitable for web app development than many other languages. One of the web frameworks written in python is Flask, which is the framework used for the toolbox. Its core is relatively small but scalable, so it may handle more complex app solutions. Flask is utilized primarily because of its user-friendliness and because it brings flexibility to application design and deployment.

Figma is used to get a rough visualization of the design and logic, but the final design realization was directly enforced in VS code and python. The proposed changes to the initial structure were easier to replicate/convert to HTML files inside the project rather than applying the changes to the Figma template and then importing the files.

Video stream

Early on, Zoom showing promise of being well suited for the application, but was canceled due to development restrictions. No permission or authorization criterion is needed for providing a video feed with the OpenCV library, and it supports integration with the python-flask framework. In combination with HTML tags, the library can be used to take screenshots and/or capture the video frame of the feed so that recordings and images can be downloaded or displayed on a web page. Other computer vision or video recording services are available, but openCV seemed the most convenient for this project. Utilizing this library makes it possible to expand the video recording features using machine learning algorithms (Face detection, eye- and object tracking, ...), enhancing the quality assessments even further.

Game engine

Unity and Unreal Engine are two well-known game engine platforms on the market and are popular among game designers and developers. Unreal is often preferred over Unity when designing 3-D games and when higher graphical quality is needed. It is designed for independent video game development and uses C++ for game scripting. Unity is created with fewer system features and uses C# as its programming language, making it easier to use when designing games with low complexity. The toolbox's analytical input only comprises the number of pressed keys on a keyboard collected during a 2-D game. Unity is, therefore, the most reasonable engine to use as the game analytics provider for the web application.

Survey SW

Questionnaire templates are shareable both in Microsoft- and Google forms. They are the most popular services used for creating surveys and/or questionnaires, where either of them is more popular than the other. They are similar in terms of costs, accessibility, and features, meaning that people choose one of them based on their preferences. The intention is to include both services in the toolbox, letting the user decide which platform to use.

4.2.4 Implementation

The results reflect on the application implementation and how users are intended to utilize the toolbox. All application pages are displayed as well as the logic behind their design and UI.

Flask in python is chosen as the application's web framework. It requires little knowledge to get started and is quite flexible regarding extension support. This will be useful when the application needs additional features. Some of the extensions used are:

- flask SQLAlchemy (Database toolkit)
- flask WTF (Adds rendering and validation to forms)

Flask extensions are used when the flask framework itself is inadequate. For more significant projects, the folder structure needs to be built up so that files can be included and used within the entire project. This will also make the code cleaner as the filename will match its content more accurately.

A minimal flask application includes a static and template folder. The template folder contains all HTML files, while the static folder contains assets that can be used by the templates (CSS, video, and JavaScript files). The actual app is defined and initiated within the app folder with configurations given in the config.py file. All database models and forms are placed in models.py and forms.py, respectively. The models are created with a so-called "one-to-many" relationship(see Figure 8 and Figure 9), which allows one user to have multiple experiments, one experiment to have many evaluation tools, and lastly one tool to have many data sets.

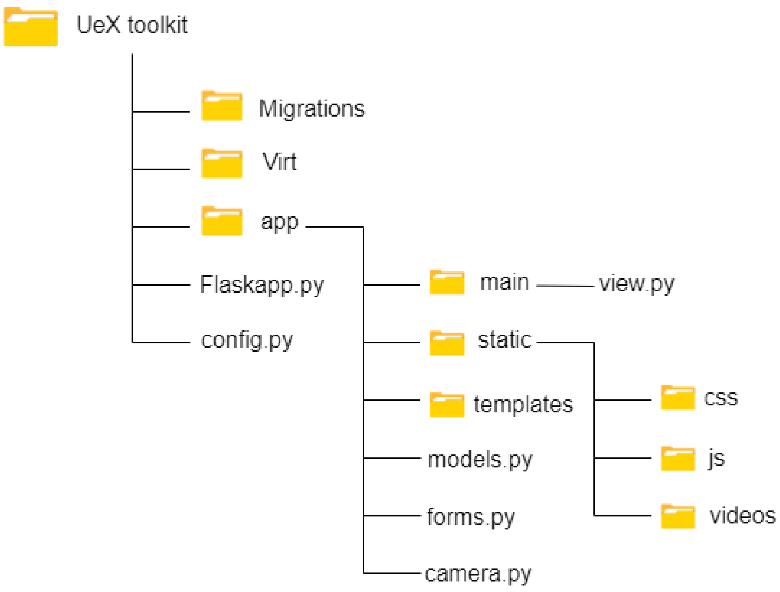


Figure 7: Project folder structure

The web framework also needs a migration folder for database editing and restructuring. The migrations folder contains the data tables' definitions and each column's name and type. Whenever a column is changed or added, the migration is pushed into the database keeping it up to date with the latest applicable version.

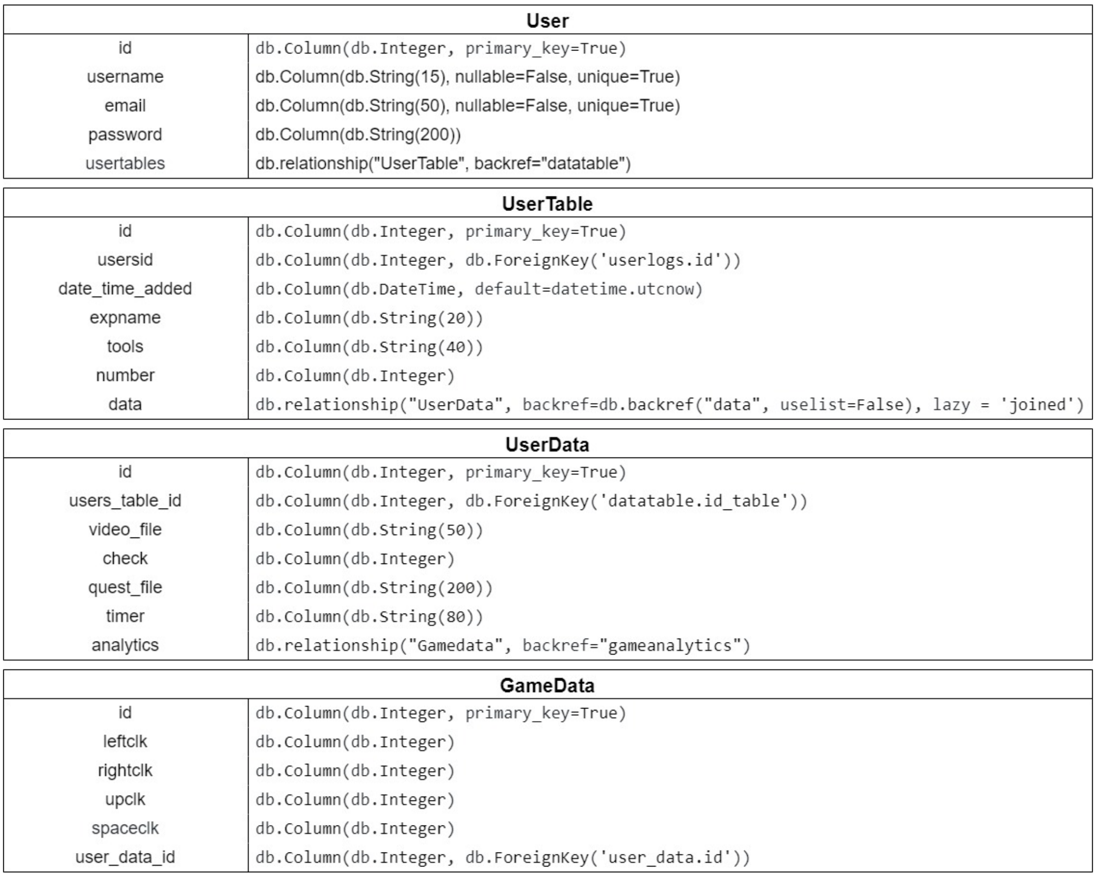
The web camera functionality is created in a separate file, camera.py, which includes setting up the recording thread and defining the video camera object logic. The definitions are being utilized in dedicated routes in the view file.

Forms are used in flask as a way to post and get data to the server as well as let the user interact with the web application. The forms are typically created in the templates and submitted to any URL route specified in the application. The flask framework handles application routing by mapping URLs to specific functions that take care of their logic. The toolbox is structured with 19 routes that either return HTML pages or handle post requests from the client. Each route is defined in the view.py file forming most of the server-side of the application.

Database storage

The toolbox uses SQLAlchemy with MySQL as a database service. The service is open source and free of charge, even for commercial use. The data stored is separated by models, one for each purpose:

- **User:** Application login data
- **UserTable:** Experiment information
- **UserData:** Experiment data
- **GameData:** Game analytics



User	
id	`db.Column(db.Integer, primary_key=True)`
username	`db.Column(db.String(15), nullable=False, unique=True)`
email	`db.Column(db.String(50), nullable=False, unique=True)`
password	`db.Column(db.String(200))`
usertables	`db.relationship("UserTable", backref="datatable")`

UserTable	
id	`db.Column(db.Integer, primary_key=True)`
usersid	`db.Column(db.Integer, db.ForeignKey('userlogs.id'))`
date_time_added	`db.Column(db.DateTime, default=datetime.utcnow)`
expname	`db.Column(db.String(20))`
tools	`db.Column(db.String(40))`
number	`db.Column(db.Integer)`
data	`db.relationship("UserData", backref=db.backref("data", uselist=False), lazy = 'joined')`

UserData	
id	`db.Column(db.Integer, primary_key=True)`
users_table_id	`db.Column(db.Integer, db.ForeignKey('datatable.id_table'))`
video_file	`db.Column(db.String(50))`
check	`db.Column(db.Integer)`
quest_file	`db.Column(db.String(200))`
timer	`db.Column(db.String(80))`
analytics	`db.relationship("Gamedata", backref="gameanalytics")`

GameData	
id	`db.Column(db.Integer, primary_key=True)`
leftclk	`db.Column(db.Integer)`
rightclk	`db.Column(db.Integer)`
upclk	`db.Column(db.Integer)`
spaceclk	`db.Column(db.Integer)`
user_data_id	`db.Column(db.Integer, db.ForeignKey('user_data.id'))`

Figure 8: Database structure

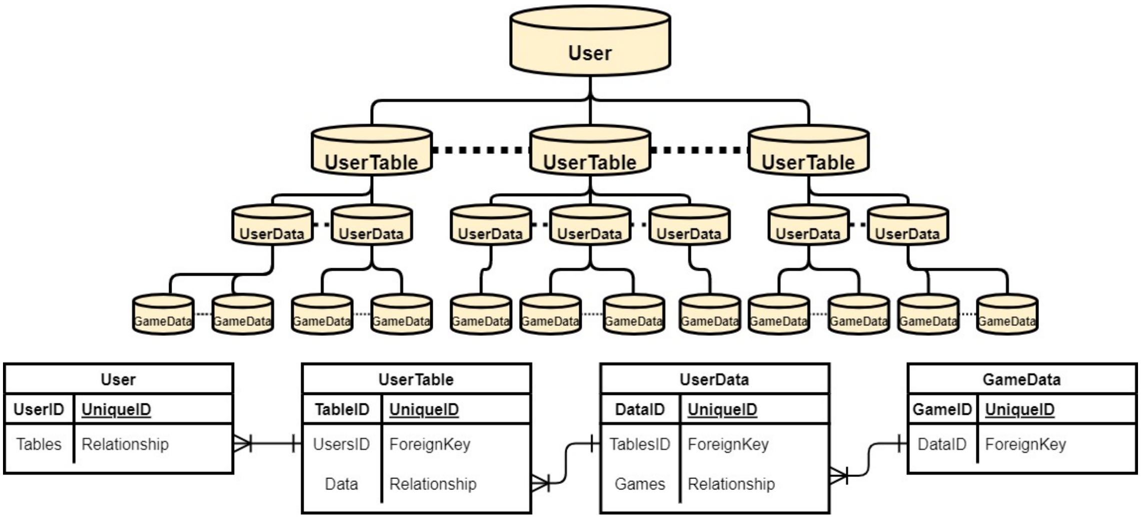


Figure 9: Illustration of one-to-many db relationship

The rows specify columns in the database and corresponding data types. The **User** model comprises the data used for application login. The first column is equal for all models, reserved for identification. When using the "primary_key" argument, the IDs will be auto-incremented as new users are added. Username and email are stored as plain text while the password is being hashed, and "salted" before it is added to the database (see Section 6). Passwords and usernames are compared with a form submitted and sent from client to server. The login and sign-up forms are created using flasks WTForms library extension to make validation and app rendering easier to handle. The **User** model is also utilizing a so-called ”one-to-many” relationship with the **UserTable** model, making it possible for one user to have multiple user tables associated with their primary key. The **UserTable** model is linked back to users with the ”ForeginKey” attribute, assigning it the same value as whatever the **User** model’s id is.

The **UserTable** model contains all information regarding an experiment, which is:

- Timestamp for when data is added(in the format => dd-mm-yy hh:mm:ss)
- Experiment name
- Experiment tools [Video recorder, Session timer, Analytics, Questionnaires]
- Number of participants

The table is also established with a relationship to the **UserData** model, as different data columns can belong to one experiment.

The remaining models created are related to the actual data collected during the experiment. The **UserData** model comprises a timer instance for a subject’s overall time spent in a session, the name of the questionnaire and scales (listed in Table 4), the name of the video file recording, and a checker used to capture the experiment progress(how many have successfully finished). The **GameData** model keeps all the analytical data obtained from playing a game created in Unity and is linked to the **UserData** model using the ”relationship-ForeginKey” fields.

The table below is intended to clarify when data is committed and added to each of the models:

Table 3: The table displays the relation between web pages and db models (Figure 8) (in which web page are data committed and added to the different db models)

Web pages and DB models	
Application page	**Database model**
SIGN UP Figure 12	User
PROFILE Figure 16	UserTable
TASKS Figure 17	UserData
GAME LIST ADD Figure 19	GameData

Application UI and design

The flowchart below is created to help you keep track of the application user flow as screenshots of each page are presented with related descriptions.

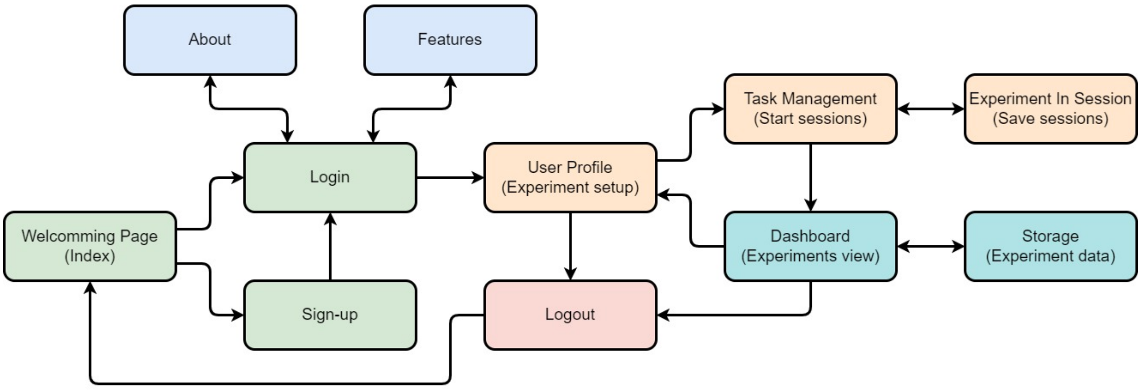
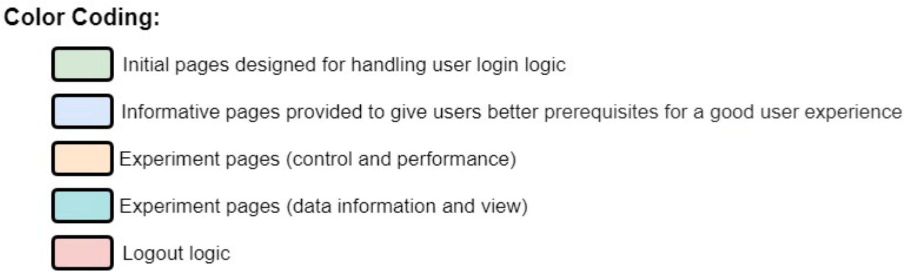


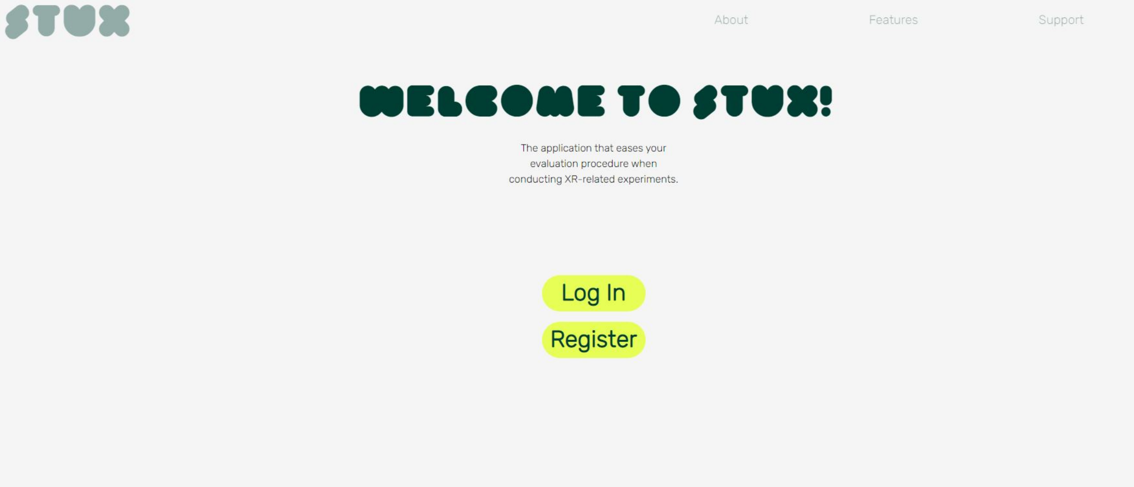
Figure 10: Application flowchart/user-flow



Color Coding:

- 🟩 Initial pages designed for handling user login logic
- 🟦 Informative pages provided to give users better prerequisites for a good user experience
- 🟧 Experiment pages (control and performance)
- 🟦 Experiment pages (data information and view)
- 🟥 Logout logic

The toolbox is made with a simplistic UI to make it easy for user interaction and utilization. As the user goes to the application URL, they are meet with the initial page displayed in Figure 11:



STUX

About Features Support

WELCOME TO STUX!

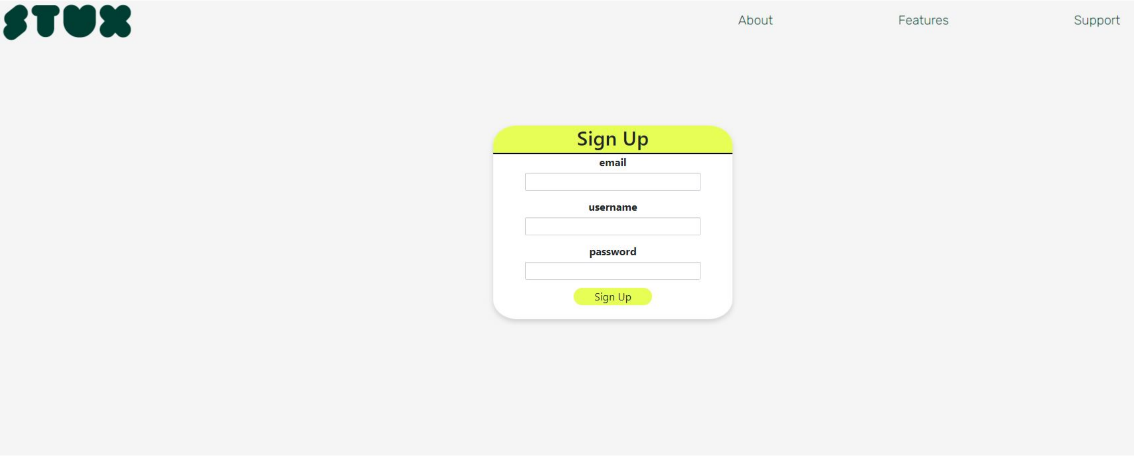
The application that eases your evaluation procedure when conducting XR-related experiments.

Log In

Register

Figure 11: Application index page

The page lets you log in or sign up for a new account. Users are required to have an account before they can utilize the application. The page is meant to introduce the app to the user to make them aware of its purpose. The ”about” and ”feature” pages provide a more detailed app description.



STUX

About Features Support

Sign Up

email

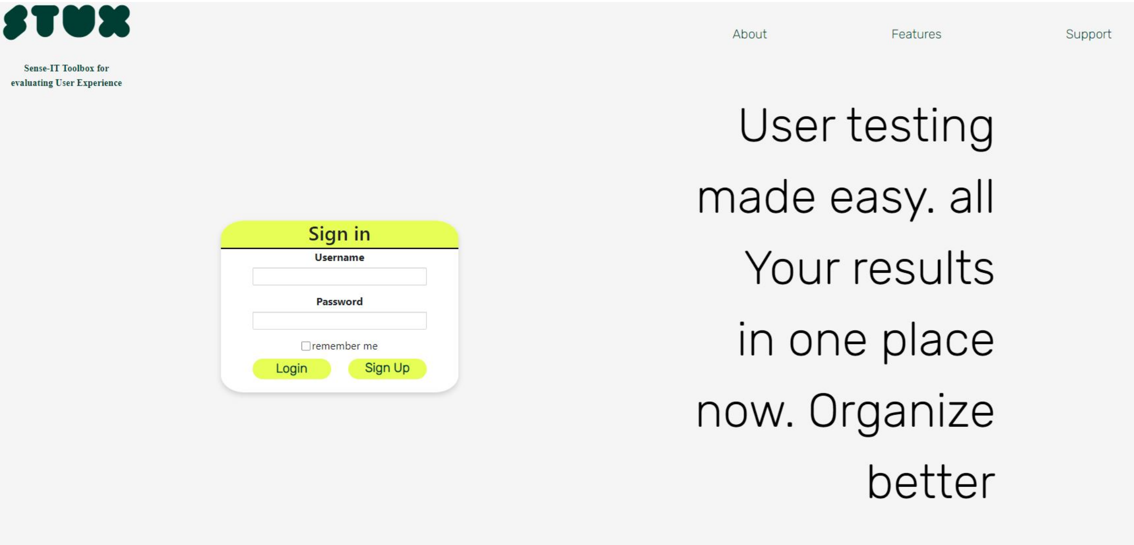
username

password

Sign Up

Figure 12: Application Sign up page

Users sign up with a username, email, and password. The username and email are unique for each account, so it is impossible to choose a name already registered. The sign-up information is stored in the application database, assigning each new user an id to make it possible to keep track of who's currently logged in. How security and data protection are handled are explained in Section 6.

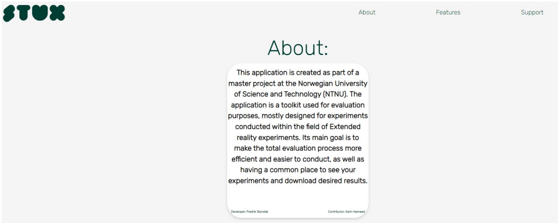


Sense-IT Toolbox for evaluating User Experience

User testing made easy. all Your results in one place now. Organize better

Figure 13: Application login page

Once the user has created an account, they can log in with their username and password. The entire database is queried and filtered by registered usernames, and the user gets access to their profile if the typed password matches the one registered. The profile page (Figure 16) is rendered once user credentials is correct.



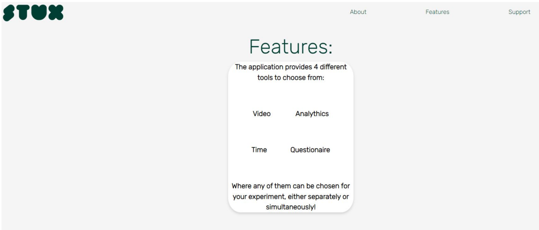
About:

This application is created as part of a master project at the Norwegian University of Science and Technology (NTNU). The application is a toolkit used for evaluation purposes, mostly designed for experiments conducted within the field of Extended reality experiments. Its main goal is to make the total evaluation process more efficient and easier to conduct, as well as having a common place to see your experiments and download desired results.

Developer: Fredrik Skorstøl

Contributor: Asim Hameed

Figure 14: Application information page



Features:

The application provides 4 different tools to choose from:

Video
Analythics
Time
Questionaire

Where any of them can be chosen for your experiment, either separately or simultaneously!

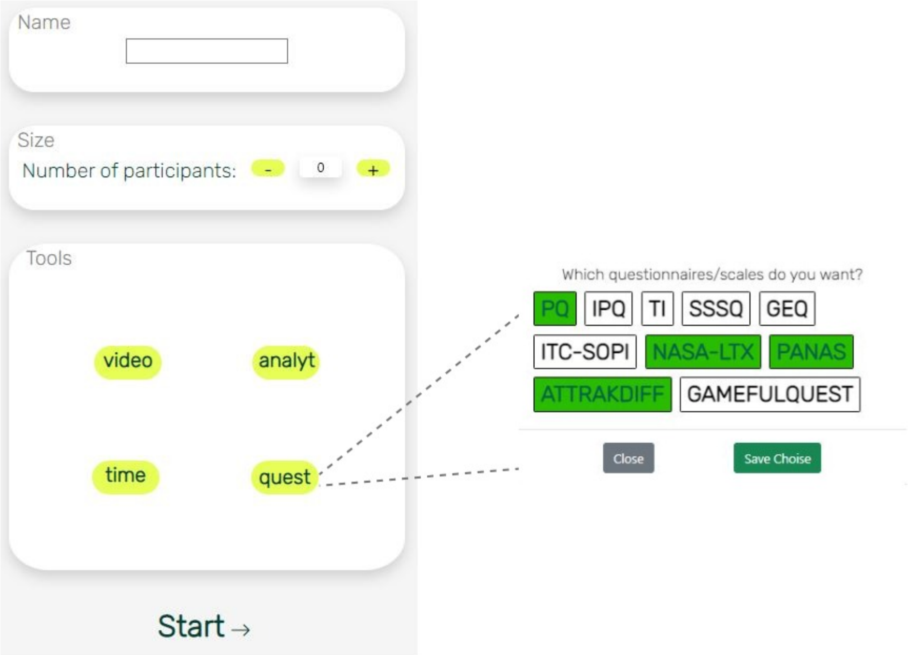
Figure 15: Application features page

Figure 14 and Figure 15 are pages that provide the user with information regarding the toolbox's:

- Origin
- Purpose
- Suitable scenarios for its use

The pages can be opened and read before signing up to ensure that the application delivers desirable service and tools. Its also intended to provide a support feature that includes FAQ and user feedback to improve application performance and uncover any discomfort or annoyance with the UI. This feature is not implemented and will be part of potential future work.

The screen in Figure 16 shows parts of a user's profile. Before starting an experiment, the user needs to give it a name, set the number of participants, and choose which evaluation tools to use. Once each field is filled out, the experiment can commence.



Name

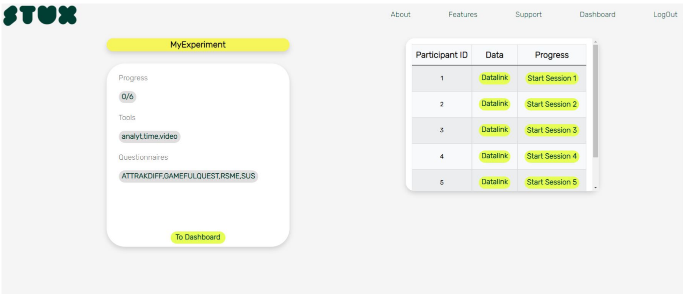
Size

Number of participants: 0

Tools

Which questionnaires/scales do you want?

Figure 16: Application profile page



MyExperiment

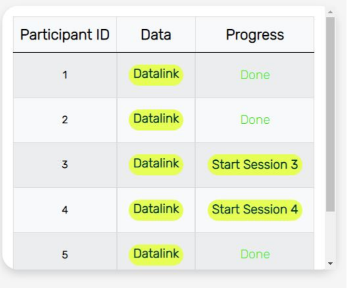
Progress: 0/6

Tools: analyt,time,video

Questionnaires: ATTRAKDIFF,GAMEFULQUEST,RSME,SUS

Participant ID	Data	Progress
1	Datalink	Start Session 1
2	Datalink	Start Session 2
3	Datalink	Start Session 3
4	Datalink	Start Session 4
5	Datalink	Start Session 5

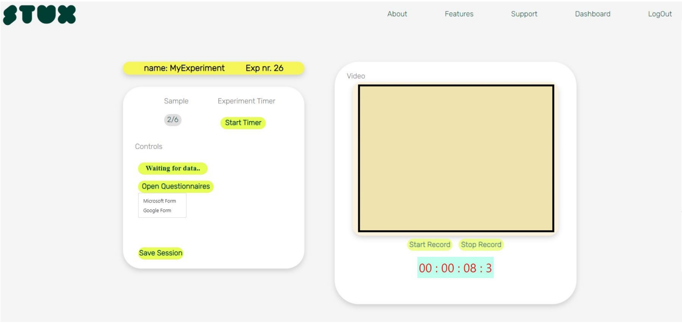
Figure 17: Starting screen for an experiment



Participant ID	Data	Progress
1	Datalink	Done
2	Datalink	Done
3	Datalink	Start Session 3
4	Datalink	Start Session 4
5	Datalink	Done

Figure 18: Progress window

All information provided will be displayed on the screen while an experiment is running. At this point, each participant is assigned an ID to keep track of the progress. Finished subjects are indicated in the progress window 18 and the data link button will render that person's collected data. The progress is also stored so the experiment can continue another day.



name: MyExperiment Exp nr. 26

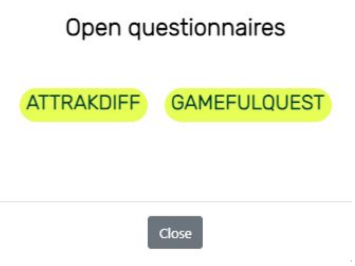
Sample: 2/6

Experiment Timer

Controls

00 : 00 : 08 : 3

Figure 19: Participant in session



Open questionnaires

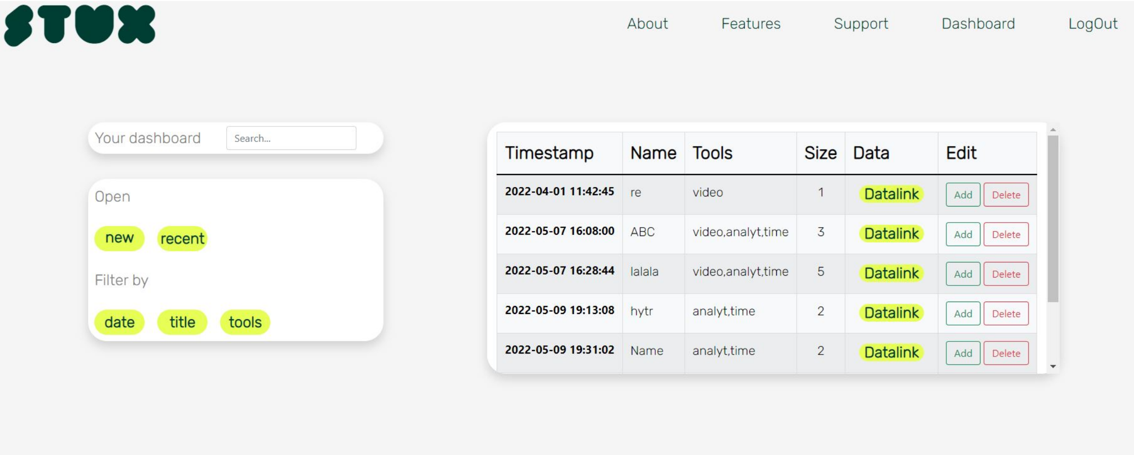
Figure 20: Open desired questionnaire

User data is collected while in session, and only the chosen tools are available for use. If a tool is excluded from the experiment, its control button is disabled. The user control includes:

- Start/Stop timer
- Start/Stop video recording
- Open questionnaire/Scale templates (Choose between Google- and Microsoft forms) and download a copy
- Start analytical data collection

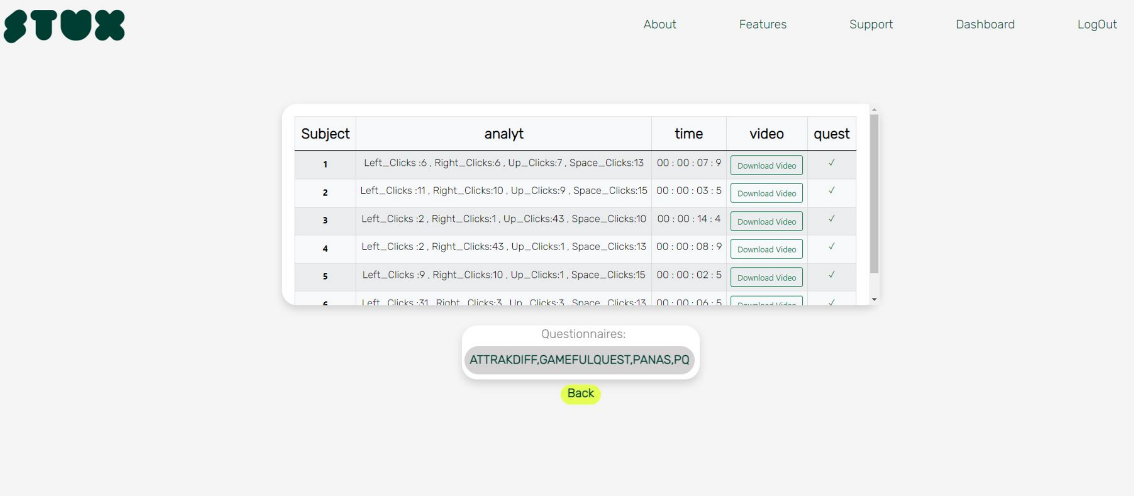
When satisfied with the result, the user can save the session and look up the result in the dashboard (Figure 22).

All experiments are located in the global dashboard page displayed in Figure 21. The user can order/filter the table by date and time it was initiated, alphabetical by title, and tools utilized. The table is also searchable by title and the three most recently conducted experiments. Each experiment is editable, so you can delete an entire experiment or add more participants.



Timestamp	Name	Tools	Size	Data	Edit
2022-04-01 11:42:45	re	video	1	Datalink	Add Delete
2022-05-07 16:08:00	ABC	video,analyt,time	3	Datalink	Add Delete
2022-05-07 16:28:44	lalala	video,analyt,time	5	Datalink	Add Delete
2022-05-09 19:13:08	hytr	analyt,time	2	Datalink	Add Delete
2022-05-09 19:31:02	Name	analyt,time	2	Datalink	Add Delete

Figure 21: Application dashboard screen



Subject	analyt	time	video	quest
1	Left_Clicks :6 , Right_Clicks:6 , Up_Clicks:7 , Space_Clicks:13	00 : 00 : 07 : 9	Download Video	✓
2	Left_Clicks :11 , Right_Clicks:10 , Up_Clicks:9 , Space_Clicks:15	00 : 00 : 03 : 5	Download Video	✓
3	Left_Clicks :2 , Right_Clicks:1 , Up_Clicks:43 , Space_Clicks:10	00 : 00 : 14 : 4	Download Video	✓
4	Left_Clicks :2 , Right_Clicks:43 , Up_Clicks:1 , Space_Clicks:13	00 : 00 : 08 : 9	Download Video	✓
5	Left_Clicks :9 , Right_Clicks:10 , Up_Clicks:1 , Space_Clicks:15	00 : 00 : 02 : 5	Download Video	✓
6	Left_Clicks :31 , Right_Clicks:3 , Up_Clicks:3 , Space_Clicks:13	00 : 00 : 06 : 5	Download Video	✓

Questionnaires:

ATTRAKDIFF,GAMEFULQUEST,PANAS,PQ

Figure 22: Dashboard for a specific experiment

The last page presents the data assembled for each experiment as an interactive table. The global timer and analytical data are stored directly, while the video can be downloaded and stored locally. You will also be able to see which questionnaires the particular experiment considers.

Video

The tool used for capturing and recording a video feed is the built-in webcam most laptops are equipped with. The software applied is the Open Source Computer Vision Library (OpenCV) which is written in C++, but has an interface with python and supports the most common operating systems on the market (Windows, Mac, Linux, Android) OpenCV_Team 2022. Choosing different video sources is available when using this library but is not considered for this application.

The video stream is, in short, implemented by following the steps:

- A camera instance is created to initialize the video and the thread responsible for video recording.
- Client send requests using "XMLHttpRequest" to start and stop video recording.
- New thread created every time an event is triggered in order to save the video stream to a file.

OpenCV defaults to raw image readings, so it is also necessary to encode the image to display the video stream on the web application correctly. The recordings are stored at the desired location using the VideoWriter function, and the final result can be downloaded from the experiment dashboard page (Figure 22)

Questionnaires and scales

There are vast variations of questionnaires and scales available when it comes to the evaluation of people's QoE. In many cases, the questions are fitted specifically for the purpose of the experiment but are often adopted from the ones universally known in the field. The application will therefore provide a collection of some well know and frequently used ones so that the users may change or adapt them as they please. The following table lists all of them with belonging descriptions.

Questionnaires	
Abbreviation	**Description**
ATTRAKDIFF:	**A Standardized User Experience Evaluation Questionnaire:** UX evaluation tool (usability & design)
GAMEFULQUEST:	**The gameful experience questionnaire:** An instrument used to measure and model the individual user's gameful experience of systems and services
GEQ:	**The game experience questionnaire:** Modular structured game experience instrument, used for measuring game experience in and after a game session
IPQ:	**The igroup presence questionnaire:** Scale used for measuring the sense of presence experienced in a virtual environment
ITC-SOPI:	**The ITC sense of presence inventory:** Tool used for measuring users thoughts and feelings experienced both after and while immersed in an environment
NASA-TLX:	**The NASA task load index:** Scale designed for obtaining workload from operators while or immediately after they are performing a task
PANAS:	**The positive and negative affect Schedule:** Scale used for measuring self-reported affect
PQ:	**The Witmer presence questionnaire:** Used for measuring presence in virtual environments
SSSQ:	**The Short Stress State Questionnaire:** Used for measuring task engagement, distress, and worry
TI:	**The temple inventory for presence:** Set of items that can be used to measure dimensions of presence

Table 4: Questionnaires and description

Each of the questionnaires is made in both Microsoft and Google forms. Their templates can be downloaded while a participant are in session(Figure 17).

Analytics

The toolbox includes some analytical data extracted from a game played and created in the Unity game engine. The game itself is cloned from a Unity repository, but the connection between the application server and the game is established, and the desired analytical data is added. The data included are keyboard inputs from the person who plays the game (the number of left, right, up, and space clicks). Post requests to the flask application are handled by a separate game script using the URI for one of the flask routes. The data is returned in the route that got the request and stored in the GameData database model.

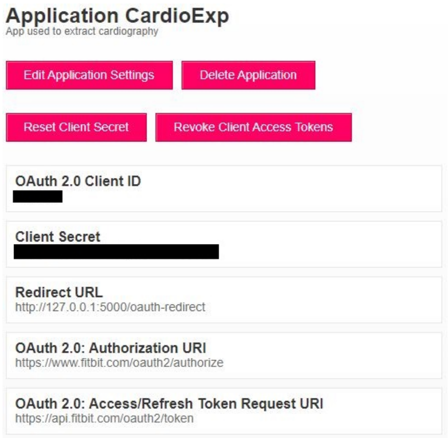
The analytical method is included to prove the concept and show that it's possible to add a bridge between the application and a Unity game. This part of the implementation is currently only suited for this particular game. Further work should therefore introduce a more general solution to make this feature meaningful for any experiment conducted.

Cardiography

Cartographic data are often collected in terms of the subjects' muscle activity, heart rate, or EDA (also known as galvanic skin response) in order to understand the person's stress level or effort put into different tasks. This kind of data and approaches were further studied in Skorstoel 2021 where only heart rate measurements are included in the prototype. The commercially available Fitbit was chosen due to its familiarity with users and because it is already owned by many. Fitbit also provides an API with python, so it should be implementable within the prototype application structure.

The Fitbit API is implemented in python and uses OAuth for authentication so that python applications may collect and read the data from Fitbit user profiles. The API documentation can be found at sphinx-quickstart 2022.

The first step toward getting the Fitbit data is registering an app at dev.fitbit.com. The registration form is filled out according to desired functionality and connection to the python-flask application. After submitting the registration form, a client ID and secret are generated. They will be used in the flask app to exchange them with an access token and refresh token, which is used to grant access to data from your Fitbit account.



Application CardioExp
App used to extract cardiography

OAuth 2.0 Client ID
[redacted]

Client Secret
[redacted]

Redirect URL
http://127.0.0.1:5000/oauth-redirect

OAuth 2.0: Authorization URI
https://www.fitbit.com/oauth2/authorize

OAuth 2.0: Access/Refresh Token Request URI
https://api.fitbit.com/oauth2/token

Figure 23: Fitbit development registration information

Once the app is authorized, heart-rate data is extracted using Fitbit API functions. The available functions in the API can be used to request the data in terms of time series within a period of 24 hours. The time span can be given as a range (from date - to date) or day (today, yesterday, ...) with a specified detail level (per sec. or min.). Unfortunately, The API does not have support for getting real-time/current heart rate values.

Another approach is to use a Fitbit SDK toolchain (@fitbit/sdk) node package available using npm package manager. The SDK is already available in Fitbit Studio, a web-based development environment requiring a user account to create an application. The watch also needs to be connected to the Fitbit mobile app so that both your phone and device can be selected in the environment.

Importing the heart rate sensor is first tested in Fitbit Studio to ensure that the modules are importable and that the phone and watch are connectable and can be selected(Figure 24). The images below show the Fitbit Studio's connection and the printed output from the current heart rate collected by the sensor.

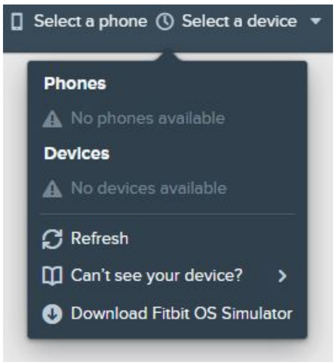
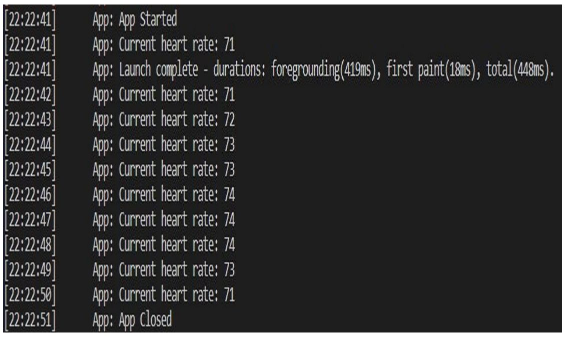


Figure 24: Connecting phone and device to Fitbit Studio



```
[22:22:41]      App: App Started
[22:22:41]      App: Current heart rate: 71
[22:22:41]      App: Launch complete - durations: foregrounding(419ms), first paint(18ms), total(448ms).
[22:22:42]      App: Current heart rate: 71
[22:22:43]      App: Current heart rate: 72
[22:22:44]      App: Current heart rate: 73
[22:22:45]      App: Current heart rate: 73
[22:22:46]      App: Current heart rate: 74
[22:22:47]      App: Current heart rate: 74
[22:22:48]      App: Current heart rate: 74
[22:22:49]      App: Current heart rate: 73
[22:22:50]      App: Current heart rate: 71
[22:22:51]      App: App Closed
```

Figure 25: Real-time HR measurements using HeartRateSensor module

After concluding that the application has connected to the device's interface and it is able to collect real-time data, the toolchain is installed and tested in a new Fitbit CLI project created in visual studio code. The example created in Fitbit Studio is copied and built in the new Fitbit CLI project. In addition to the Fitbit SDK node package, sdk-types are added in order to work with TypeScript modules (from CLI run: npx fitbit-sdk-types). The heart rate sensor is importable with all the packages installed, and the measured values are viewed in the console log (Figure 25). The next step is to combine the Fitbit CLI project with the toolbox web app. The prototype project needs to be reconstructed, and the npm initialized such that flask can serve node packages. The packages are not included in the project as flask is not able to serve them as they should, so cardiographic data are excluded from the toolbox.

4.3 Troubleshooting

Issues emerge both during the implementation and deployment of the application. Since so many different frameworks and development kits need to be integrated, finding a solution that will work for them all is not easy. Including the cardiographic data from a Fitbit has proven to be especially challenging, as the Fitbit API developed for python does not provide enough functionality for measuring real-time heart rate. Establishing a connection between the toolbox interface and the Fitbit watch is also a tedious process because it requires authentication.

Form post- and get requests have also caused a few problems. When many user inputs need to be sent from client to server at once, there are issues with packets not being sent. Different post/get methods require different venison of the libraries to enable the exchange of JSON formatted content. There were additional complications when the application was deployed to Heroku (A cloud-based platform) instead of running the application service at a local instance(change flask environment from development to production). The deployment caused errors in the toolbox functionality where no issues were uncovered while running the application locally.

The video feed stopped working after it was deployed to Heroku. The reason is that the cloud server cannot access the web camera the same way it is by running the application locally. The explored solution was to change to socket connection for the video feed, which is accomplishable with one of the flasks extensions(Flask-SocketIO. I succeeded in accessing the video feed in the Heroku app but could not get the video recording functionality to work. The video stream also caused problems with posting the other data that were to be collected in the session.

Developing a video recorder using the Zoom SDK was not possible as it required additional privileges for my zoom account. I talked to the universities IT people to figure out how to gain access, but they could not grant the proper rights. There will always be issues when using third-party software for personal development, especially regarding authentication or validation requirements.

5 Evaluation of toolbox UI

Testing and validating an application's usability and performance is essential to comprehend better its different parts and potential issues with design and UI. Getting others' opinions and using their feedback will help decrease problems and bottlenecks, which often are not that evident to the developer. The usability of an application is often tested so that others can use the reported experiences for further development and make improvements.

The toolbox is evaluated by design and UI in terms of feedback from test subjects. The SUS will be applied as rating scale, which includes ten claims, each answered with a 5-point grading (from ”strongly disagree” to ”strongly agree”). The evaluation is based on feedback from 10 people, all with engineering backgrounds. Only one of them is aware of how evaluation procedures are conducted, while the rest have been given a short introduction to the subject before testing the application. Once each participant has started the test, they are not given further instructions.

Each question are scored from 1 to 5 points. The results are converted to a 0-100 point scale by:

- Summation of all odd-numbered questions subtracted by 5
- 25 subtracted by all even-numbered questions
- Summation of all converted results multiplied by 2.5

and the usability of the application can then be graded according to the total score as:

Score	Adjectively Grading
$\geq$ 80.3	Excelent
68 - 80.3	Good
68	Ok
51 - 68	Poor
0 - 51	Awful

Table 5: SUS grading

The final result obtained by using this method is calculated to be = **52.5**, giving the application a poor usability performance rating. The toolbox is according to the test group a bit hard to follow and the UI not that intuitive. Most of the questions have the majority of answers scored by 3 points, indicating that they are quite neutral to the claims. The results are further discussed in section Section 7

The questionnaire is sorted into even- and odd-number questions, Figure 26 and Figure 27, where the odd-numbered gives higher score if the participants strongly agree (5) with the statement. Even-numbered questions gives a higher score if participants strongly disagree (1) with the claim.

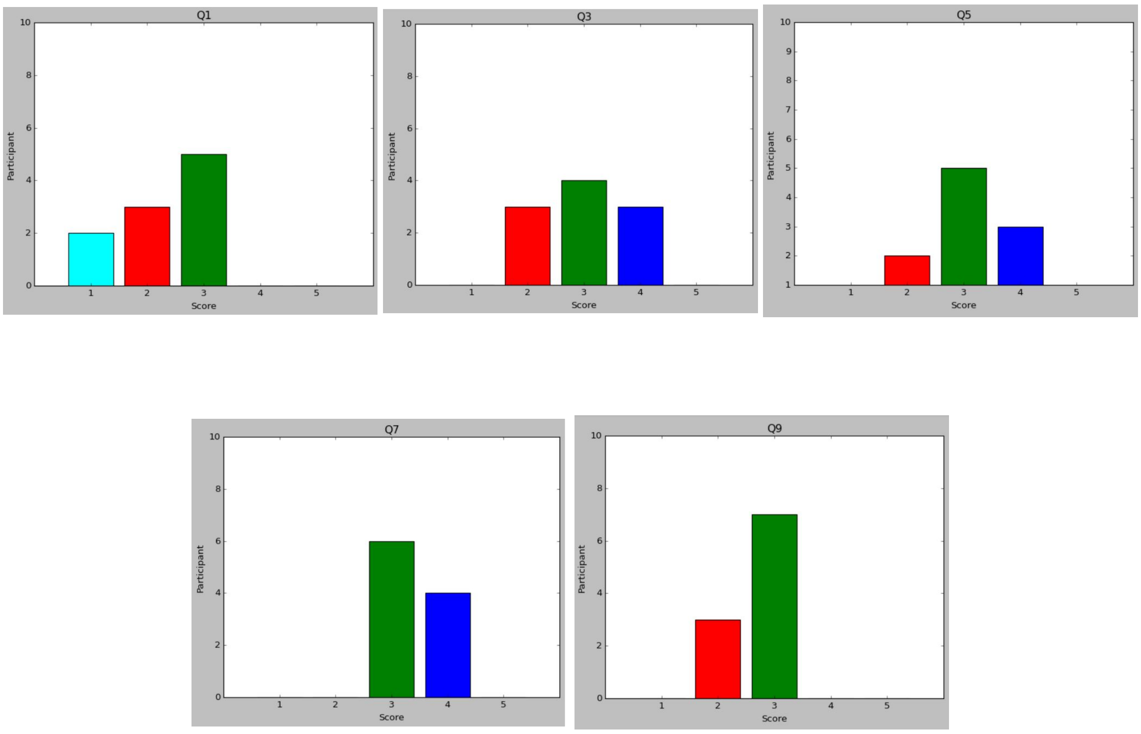


Figure 26: Odd-numbered questionnaires

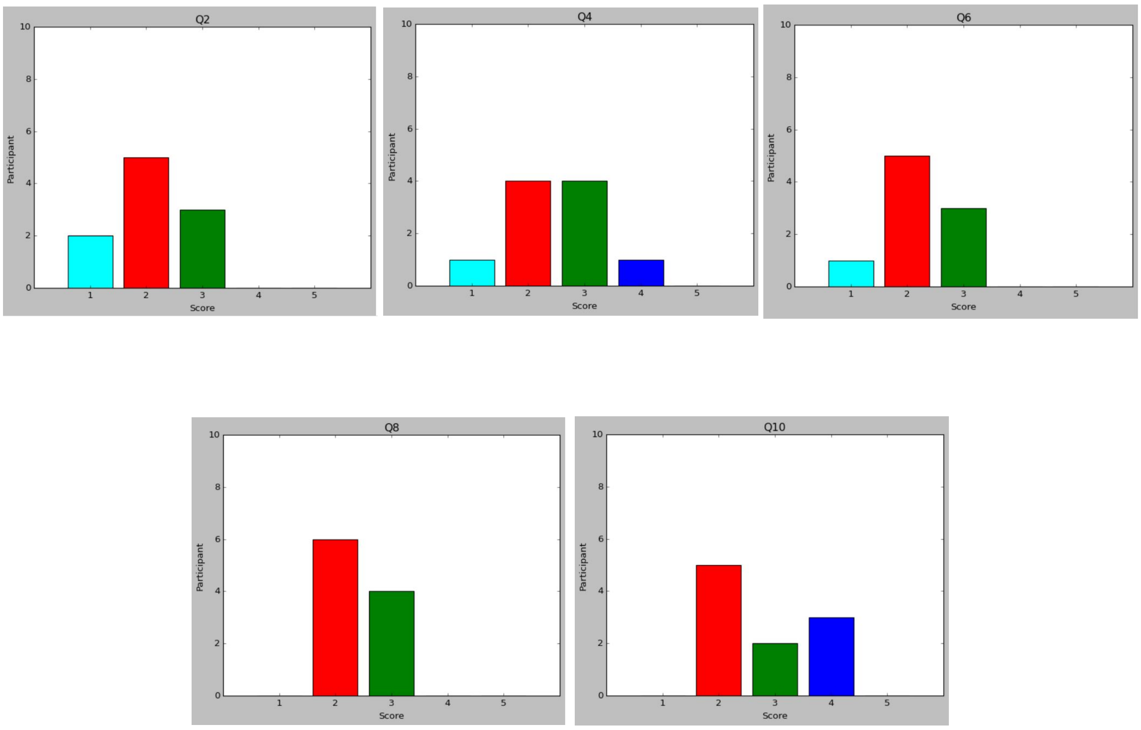


Figure 27: Even-numbered questionnaires

The results are not as good as hoped since the intention was to create an easily manageable application, and the user could navigate through its different parts without difficulties. The score may indicate otherwise because most participants experienced the UI as a bit complex and struggled to follow the procedure from experiment start to finish. Multiple possible reasons could explain the low performance score, but it is most likely due to the fact that 9 out of 10 people have low conditions for understanding the evaluation procedure. From the results, we see that most of them are not interested in using the application frequently and would require more knowledge to utilize it correctly. The SUS scale should be adopted to fit better the people involved in the usability study so that the score solely reflects design appearance and application interface. Another reason for obtaining a low SUS score is the designed user inputs. The functionality for each button or text field should be more explicit, either by including a more thorough user guide in the about/feature pages or by having more intuitive texts linked to the different inputs.

6 POPD and ethics

Ethical aspects are essential to consider when developing an application involving human subject participation. Ethical issues may occur in experiments where people's privacy is violated or questionable activities are engaged. This topic is thoroughly discussed in Perkis 2021a, particularly aimed toward experiments conducted in the mixed reality framework. The deliverable presents specific procedures followed before participants can engage in activities and identifies potential situations that may provoke ethical issues. The first step towards starting an experiment, especially those involving human participation, is an approval of a Research Protocol, which considers the experiment's ethical requirements and privacy-related issues. After the research protocol is approved by the team leader and ethical issues are uncovered, it may be filed for further approval by an ethics committee. If no ethical considerations need to be accounted for, the test can be registered, and the recruiting process can begin. The recruiting strategy depends on activities involved in the experiment and the subject's earlier experiences and backgrounds. Once the participants are selected, they receive information about the scope and aims of the project so that everything is clear before any activities start. The last step in the procedure is to give the participants a consent form, which needs to be signed before taking part in the assessment. This form will include information in conjunction with the project content, the expected duration of the different activities, the possibility of withdrawing from participation, and risks the study might involve.

Protection of Personal Data (POPD) are another segment that needs consideration. When gathering data from experiments involving human participation, it is required to do so in accordance to some regulations. Perkis 2021b describes how data collected from people and surveillance of their behavior can be handled for projects within the mixed reality framework. It follows the General Data Protection Regulation (GDPR) which describes data subjects rights and the obligations a data controller and possessor may have. Following measures can be taken as precaution towards security and safe storage:

- Data is used only for the project itself, so no processing of sensitive data is required.
- Protect data through appropriate measures and delete it after it has served its purpose or by request from the data subjects.
- Ensure that data storage is secure and that deletion is irreversible
- minimize the data collection and processing so that the content is only strictly necessary for running an activity.

6.1 Storing user login information

You need to sign up with a username, mail and password in order to use the app. Once you are signed up, you can log in and get redirected to your profile, which only will be available if the password matches the username. The application stores the user name and email as they are, but storing the password is more risky. The approach used to overcome this security issue is to hash the password before storage. That means the string typed into the password input field is transformed into a scrambled version of itself and there is no way to convert it back to its original formWerkzeug 2022. In addition to hashing, the password is ”salted” which adds an unique random string only known by the site in front of the password hash. This will increase security even more, protecting against certain attack methods.

6.2 Participant data

In order to keep the identity of the persons that are attending an experiment, none of their personal information is stored in the application when setting up a new test. The head of the experiment will only be able choose the number of subjects that are participating, and the app will store an unique id for each of them. Information about the subjects name, gender, and age is up to the user of the toolbox to keep track of and link to their assigned id's.

Test subjects answers from questionnaires are also important to secure so that data protection is maintained. This issue is avoided as the toolbox does not store any ratings submitted by the participants. The application are instead providing templates for all questionnaires and scales so the user can download them and adapt them as they please.

7 Discussion

In this study a prototype toolbox for user experience evaluation is presented and developed. Every step of the prototyping process are provided, including user feedback on the applications UI. The project builds upon the prestudy conducted last semester to establish the theoretical foundation and exploring different immersive media evaluation techniques. The final resulting toolbox comprise 4 different evaluation tools, which is; a global experiment session timer, Video stream with recording capabilities, experimental analytical data collection, and a number of standardized questionnaires and scales. Decisions regarding what type of software/framework/equipment to use are based on performance and their provided interface so that every component of the prototype connects and delivers one complete solution.

Working on multiple platforms and software, especially for those that requires third-party software and special user privileges, is not an easy task. They might work fine just by them selves, but when integrated with other tools and programs, problems may emerge. That is why open source programs are preferable, where you have more freedom during development as they often are more compatible with other software tools and programming languages.

The toolbox developed is a prototype of the concept, so there is room for further improvements to strengthen the solution's robustness and functionality. To make the application more versatile and relevant for a broader audience, it should provide an increased number of evaluation tools and features. Based on what people use regarding QoE evaluations and IMEx assessments, the toolbox would become more appealing if some physiological measures were implemented. It is relatively common to use heart rate and muscle activity measurements to evaluate a system or product's performance towards immersiveness since data is sampled without any abruptions.

It is also worth mentioning that there is not a lot of equipment available at the lab which may be used for this project. The only appropriate equipment is a Fitbit pulse watch, which seems most desirable for the heart rate measurement approach. As a recommendation, some equipment or device that measures the participant's EDA should be acquired since it has been applied in numerous evaluation studies. Most new smartwatches integrate such sensors but require the participant to place their palm or fingers on the watch and keep it there for a considerable duration, affecting their performance during experiment testing. A dedicated device to measure the EDA would, therefore, be more efficient. Measuring muscle activity is also quite common but requires more than one simple device. Some EMG probe solutions are affordable and commercially available, but the technology is still primarily used in hospital settings and for medical purposes. Other evaluation measures are easier to capture and are often obtained using downloadable software solutions (questionnaires, video recording, event tracking, game analytics, and so on).

Combining the CLI created Fitbit code with the application has proven to be challenging and quiet time consuming, as the heart-rate module can not be directly included in the application scripts. It was attempted to include python webpack, a npm module used for bundling JavaScript files, so that the browser recognize the heart rate package. The package installation caused errors and were incompatible with the rest of the applications virtual environment. A possible cause to this problem is the structure of the application folders, meaning that the folder structure needs to be changed in order for the toolkits web framework to serve node packages. This will take a lot of time, without any assurance that it works, so the two separate working projects have not been merged at this point.

Almost every study relies on user ratings, either through surveys, interviews, or audio recordings. The toolbox should therefore include a broader selection of user feedback procedures. Some studies also use more advanced equipment, such as ECG caps/probes, to measure the participant's brain activity during tests. Since products with ECG technology are not exactly commercially available, the method should not be a priority for future work. The toolbox's performance and usability on the other hand needs furter examination.

The feedback obtained by the conducted user test showed that the SUS rating result was below average, meaning the application's usability can be graded as poor and over-complicated. Most participants found it hard to follow the toolbox logic from experiment start to finish. They struggled mainly with data collection within a session and how to interpret the results. It should be taken into consideration that only one of them has actually conducted immersive media-related experiments before, so the results may be affected by confusion about the procedure to follow. It is also possible that the questions in the SUS themselves were too irrelevant for most of them. We can see that the majority did not think they would use the application frequently and felt they needed to learn more before considering using it. The necessary steps toward fixing this issue would be to improve the toolbox prototype structure and performance and make the feedback more reliable and accurate by repeating the evaluation study with more people experienced within the field.

Conclusion

This paper presents a web-based application toolbox design and development for user experience evaluation. The toolbox provides some of the many methods used to evaluate experiments within VR and other immersive media content. It is meant to make evaluation procedures more efficient and easier to manage. Most of the theoretical research is based on earlier conducted experiments, prototypes, and novel designs. The research increase the knowledge of how experiments are conducted, which activities they comprise, and the type of technology or methods used for their evaluation. It is used to decide which tools to include in the application by implementing the ones most commonly used.

The information is further analyzed to give a broader picture of the many possibilities you may have for evaluation, considering the methodology and the data formats they provide. The application utilizes the Flask framework with feature extensions to get the desired functionality. The tools provided are a timer, video recorder, analytical collector, and questionnaire templates, each with a different set of requirements for implementation. The analytical feature provided is only included to prove the concept and show that it is possible to establish a link between a game created in Unity and the application. The intention was also to incorporate a cardio graphical tool to provide the user with heart rate data. The tool was dropped since the python-Fitbit API currently does not provide functionality for obtaining the real-time heart rate status.

Data protection and ethics are addressed due to the application's nature. Protecting data is especially important in experiments where humans are conducting activities, and the data collected can be misused or subject to privacy violations. The toolbox is implemented with concern for these issues by securing the data with login credentials and not storing any personal information.

The final result is an application that manages all your experiments, lets the user decide how many people are involved, and gives you four different tools. Data the tools provide is collected during experiment sessions which can be viewed in the application dashboard for further analysis. Video recordings and questionnaire answers will not be displayed in the web application in order to maintain data protection. You will instead be able to download the videos for each participant and get the questionnaire templates so they can be stored by the experiment conductor locally on their computer.

There are still some logical issues regarding the data collection and posts requests sent from application client to server, making the UI less intuitive than intended. The UI is according to the feedback obtained from the SUS quite cumbersome and should therefore be considered to be redesigned to give the user a better experience.

Bibliography

[1] P. Milgram, H. Takemura *et al.*, *Augmented reality: A class of displays on the reality-virtuality continuum*, 1995. DOI: 10.1117/12.197321. [Online]. Available: https://doi.org/10.1117/12.197321.

[2] A. Perkis, C. Timmerer *et al.*, *Qualinet white paper on definitions of immersive media experience (imex)*, European Network on Quality of Experience in Multimedia Systems and Services, 14th QUALINET meeting (online), May 2020. [Online]. Available: https://arxiv.org/abs/2007.07032.

[3] K. Brunnström, S. A. Beker *et al.*, *Qualinet white paper on definitions of quality of experience*, 2013. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00977812.

[4] B. G. Witmer and M. J. Singer, *Measuring presence in virtual environments: A presence questionnaire*, 1998. [Online]. Available: https://doi.org/10.1162/105474698565686 (visited on 28th Oct. 2021).

[5] K. Bektaş, T. Thrash *et al.*, *The systematic evaluation of an embodied control interface for virtual reality*, 2021. [Online]. Available: https://doi.org/10.1371/journal.pone.0259977.

[6] R. Schatz, A. Sackl *et al.*, *Towards subjective quality of experience assessment for omnidirectional video streaming*, 2017. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7965657.

[7] F. De Simone, J. Li *et al.*, 'Watching videos together in social virtual reality: An experimental study on user's qoe', in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2019, pp. 890–891. DOI: 10.1109/VR.2019.8798264.

[8] D. Tsetserukou, K. Sato and S. Tachi, *Exointerfaces: Novel exosceleton haptic interfaces for virtual reality, augmented sport and rehabilitation*, 2010. DOI: 10.1145/1785455.1785456. [Online]. Available: https://doi.org/10.1145/1785455.1785456.

[9] S. Scheggi, L. Meli *et al.*, *Touch the virtual reality: Using the leap motion controller for hand tracking and wearable tactile devices for immersive haptic rendering*, 2015. DOI: 10.1145/2787626.2792651. [Online]. Available: https://doi.org/10.1145/2787626.2792651.

[10] C. Boletsis and J. E. Cedergren, *Vr locomotion in the new era of virtual reality: An empirical comparison of prevalent techniques*, 2019. [Online]. Available: https://downloads.hindawi.com/journals/ahci/2019/7420781.pdf.

[11] T. Kojic, U. Sirotina *et al.*, *Influence of ui complexity and positioning on user experience during vr exergames*, 2019. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8743273.

[12] F. Skorstoel, 'Immersive media experience evaluation toolkit', Department of Electronic Systems, NTNU – Norwegian University of Science and Technology, Project report in TFE4590, Dec. 2021.

[13] C. Keighrey, R. Flynn *et al.*, 'A qoe evaluation of immersive augmented and virtual reality speech & language assessment applications', in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, IEEE, 2017, pp. 1–6.

[14] M. S. Anwar, J. Wang *et al.*, *Measuring quality of experience for 360-degree videos in virtual reality*, 2020. [Online]. Available: https://doi.org/10.1007/s11432-019-2734-y.

[15] S. Katsigiannis, R. Willis and N. Ramzan, 'A qoe and simulator sickness evaluation of a smart-exercise-bike virtual reality system via user feedback and physiological signals', *IEEE Transactions on Consumer Electronics*, vol. 65, no. 1, pp. 119–127, 2018.

[16] Fitbit, *Web api reference*, 2022. [Online]. Available: https://dev.fitbit.com/build/reference/web-api/.

[17] Thryve, *Integrate apple healthkit with your app*, 2021. [Online]. Available: https://thryve.health/apple-health-kit-integration/.

[18] Zoom-Video-Communications, *App marketplace*, 2022. [Online]. Available: https://marketplace.zoom.us.

[19] Epic-Games, *In-game analytics*, 2022. [Online]. Available: https://docs.unrealengine.com/4.27/en-US/TestingAndOptimization/Analytics/.

[20] OpenCV_Team, *About*, 2022. [Online]. Available: https://opencv.org/about/.

[21] sphinx-quickstart, *Fitbit api*, 2022. [Online]. Available: https://python-fitbit.readthedocs.io/en/latest/.

[22] A. Perkis, *D5.5 ethics issues related to human subject participation*, 2021. [Online]. Available: http://www.admire3d.eu/data/.

[23] ——, *D5.6 popd*, 2021. [Online]. Available: http://www.admire3d.eu/data/.

[24] Werkzeug, *Security helpers*, 2022. [Online]. Available: https://werkzeug.palletsprojects.com/en/2.1.x/utils/.

A Program

A.1 Github repository

https://github.com/Fredrik96/uex-app.git