Interaction design for automatic tracking of free weight exercises using computer vision and IoT systems

Axel Mulder

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2022

MASTER'S THESIS

SONY

Interaction design for automatic tracking of free weight exercises using computer vision and IoT systems

Axel Mulder
dat12amu@student.lu.se

June 13, 2022

Master's thesis work carried out at Sony Europe B.V..

Supervisors: Jakob Håkansson, jakob.hakansson@sony.com
Günter Alce, gunter.alce@design.lth.se

Examiner: Joakim Eriksson, joakim.eriksson@design.lth.se

Interaction design for automatic tracking of free weight exercises using computer vision and IoT systems

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: Jakob Håkansson, Günter Alce
Examiner: Joakim Eriksson

Abstract

Computer vision is on the cusp of revolutionizing several different markets through its applications. One such market is that of fitness and exercise. Advagym, Sony's connected gym solution is currently employing IoT solutions to track exercises in a gym environment. Together with Sony's R&D Center Lund Laboratory's computer vision- and machine learning-equipped tracking system - they are exploring solutions to free weight exercise tracking.

Using a human-centered design process - several iterations of prototypes with varying degrees of fidelity were developed, in order to find and evaluate a solution that would enable Advagym to offer an automatic free weight exercise tracking solution to its users. The final iteration of the prototypes was implemented as a fully-functional smartphone application running on the iOS and Android operating systems. The application leveraged both Advagym's IoT devices and the Sony's R&D Center Lund Laboratory's tracking system in order to deliver on its purpose.

Four final prototypes were put through a usability test with 20 participants in order to evaluate several modes of interaction and user interfaces that enabled the automatic free weight exercise tracking. The test showed that the prototype had an above average user experience - and 90% of the test participants would like to use a system such as the one proposed, if it existed on the market.

Keywords: Computer Vision, Interaction Design, IoT, Machine Learning, User Experience, User Interface

Sammanfattning

Datorseende är på väg att revolutionera flera olika marknader genom dess applikationer. Detta inkluderar marknaderna fitness och träning. Advagym, Sonys uppkopplade gymlösning använder för närvarande IoT-lösningar för att track:a övningar i en gymmiljö. Tillsammans med Sonys R&D Center Lund Laboratorys datorseende- och maskininlärning-utrustade tracking system - utforskar de lösningar för tracking av friviktsövningar.

Med hjälp av en människo-centrerad designprocess togs det fram flera iterationer av prototyper med varierande utvecklingsgrad, för att hitta och utvärdera en lösning som skulle göra det möjligt för Advagym att erbjuda en lösning för automatisk tracking av friviktsövningar till sina användare. Den sista iterationen av prototyperna implementerades som en fullt fungerande smartphoneapplikation som körs på iOS- och Android-operativsystemen. Applikationen utnyttjade både Advagyms IoT-enheter och Sonys R&D Center Lund Laboratorys tracking system för att kunna funktionera.

De fyra sista prototyperna var en del av ett användbarhetstest med 20 deltagare - där de olika interaktionssätten och användargränssnitten som möjliggjorde automatisk spårning av friviktsövningar, utvärderades. Testet visade att prototypen hade en användarupplevelse som var över genomsnittet - och 90% av testdeltagarna skulle vilja använda det föreslagna systemet, om det fanns på marknaden.

Nyckelord: Användargränssnitt, Användarupplevelse, Datorseende, Interaktionsdesign, IoT, Maskininlärning

Acknowledgements

I would like to thank Advagym for hosting this thesis work. Thank you to Henrik Bengtsson for the opportunity of working with you and your team and for taking an interest in my work and providing feedback. Thank you to all Advagym team members for the lunch discussions about all things Advagym and more - you made me feel like a part of the team.

I would also like to thank the Sony's R&D Center Lund Laboratory team; Andrej Petef and Sangxia Huang for the opportunity to work with them, along with their patience, understanding and guidance. A special thank you to Adam Jalkemo for all of the in-depth discussions, encouragement and thoughtful help provided.

Another thank you goes to my supervisor at LTH, Günter Alce, who has guided me through the entire thesis process from its inception and for the feedback, enthusiastic interest and support provided along the way.

Lastly, I would like to thank my Advagym supervisor, Jakob Håkansson, for his camaraderie, expertise, enthusiasm, time and energy spent during the entirety of this thesis work. You have taught me a lot about what goes into the engineering process - but also a lot about myself. Thank you for this.

Lund, May 2022

Axel Mulder

List of acronyms and abbreviations

CTS - Camera-based Tracking System

MQTT - Message Queuing Telemetry Transport

IoT - Internet of Things

AI - Artificial Intelligence

ML - Machine Learning

Lo-Fi - Low Fidelity

Hi-Fi - High Fidelity

SUS - System Usability Scale

UI - User interface

UX - User experience

API - Application Programming Interface

NFC - Near-Field Communication

BLE - Bluetooth Low Energy

ToF - Time-of-Flight

TTS - Text-to-Speech

SVG - Scalable Vector Graphics

ID - Identity Document

PT - Personal Trainer

Contents

Chapter 1

Introduction

Physical exercise is of major importance for all adults and adolescents. It vastly improves people's mental health, sleep, physical ability, weight control - but also decreases risk of many chronic diseases and ailments [1].

Therefore, it is in the interest of society writ large - both economically as well as socially - to help and encourage people to exercise and stay healthy. A growing data- and technology-based industry is currently fostering new ways to assist with this very matter.

1.1 Background

Global gym and fitness market & applications

The global gym and fitness industry has seen its market size grow by over 43% from 2009 to 2019 ($67B to $97B) [2]. Due to the COVID-19 pandemic leading to nationwide lockdowns along with social distancing regulations and norms - the transition from traditional gyms to virtual fitness has been accelerated. This has specifically led to increased downloads and subscriptions of fitness applications. The global fitness application market size is valued at $1.1B (as of 2021) - and is expected to see a compound annual growth rate (CAGR) of 17.6% from 2022 to 2030 [3].

Due to the trends and reasons stated above - there is a clear incentive for different companies and/or entities to support this growth and market expansion. As such, there are currently a vast amount of fitness applications on the market that satisfies specific niches or needs of the user. Such needs encompass everything from the monitoring, planning or logging of the user's exercises and training progress - to simply offering motivation and encouragement to the user. However, some companies are aiming to offer the user a more-or-less "complete" solution: thereby "digitizing" the entire gym-going experience.

Supporting gym-goers

The gym-going experience itself is often a complicated affair due to the vast amount of different exercises that one can perform, variations thereof, what kind of gym equipment is available at one's specific gym, what training methodology one would like to employ, etc.. As such, it would be beneficial for a potential gym-goer to be supported in various ways in order to make the gym-going experience more user friendly, intuitive and also more inviting to "newcomers". An example of such support could be in the form of letting users log and track their completed "sets" (where each "set" consists of a sequence of "repetitions" of a certain movement, pertaining to an exercise). Another example would be to guide and help beginners (as well as experienced gym-goers) through the use of tailored "programs" that contain different exercises available at one's gym.

A lot of fitness applications support the two above-mentioned examples through means of manual user input and logging. However, as mentioned above - there are a slew of different aspects/statistics that a gym-goer could be interested in tracking and logging - which makes for a fair amount of work and effort required in doing so. An intuitive way of supporting a user in this regard for an example, would be to collect exercise data (specifically sets- and repetitions-data) and logging it automatically (and digitally). This is one of many ways in which companies/entities are seeking to digitize the gym-going experience.

Advagym and the gym-going experience

Advagym is the technology company Sony's "connected gym" solution. It is currently using Internet of Things (IoT) devices with sensors, that are mounted on gym machines and equipment in order to track and log user's exercises automatically [4]. The users then interact with Advagym's software platform through the use of smartphone applications. Advagym's software platform also enables gyms to offer their members tailor-made programs as well as Personal Trainer (PT) services.

However, Advagym currently does not support automatic tracking and logging of an important aspect of the gym-going experience: free weight exercises. Recent advances in computer vision and machine learning have presented possibilities to solve this "problem".

Sony's R&D Center Lund Laboratory has developed a Camera-based Tracking System (CTS) that enables three dimensional (3D) object-tracking and can be trained to detect specific movement patterns. Through the use of this system - Advagym and R&D Center Lund Laboratory are jointly exploring implementations of an automatic free weight exercise tracking solution. Since the CTS itself is "invisible" to the user - with no means of input or interaction - how is the user supposed to *use* the system?

Research questions

All of this leads into the research questions of this paper:

- What are the best practices in regards to designing user-facing interfaces for 3D object-tracking camera-based computer vision systems?

- How do you evaluate and analyze the robustness and efficacy of such a user interaction, with the system?
- How do you offer and visualize onboarding, feedback, user flows and different interaction modes of such a system, to a user?
- Which of the proposed approaches/solutions/prototypes is the most compelling, from a real-world business perspective (specifically for integration into Advagym's current systems)?

1.2 Scope

The scope of the thesis work is to study and evaluate various modes of interaction and user interface(s) needed for users moving around in physical space, interacting with a complex camera-based computer vision- and IoT-system.

Currently, users of the Advagym's system (software platform) interact with it by use of an application on their smartphones (running the iOS or Android operating system). By implementing a smartphone-based prototype application that can interact and communicate with Advagym's hardware (IoT devices) and Application Programming Interfaces (APIs) - along with Sony R&D Center Lund Laboratory's CTS - it will be possible to evaluate, benchmark and try different approaches in order to answer the research questions stated above.

The thesis work and project should also result in something of direct value to Advagym (and Sony's R&D Center Lund Laboratory) - in regards to what design decisions can improve their solutions and products going forward. Therefore, a couple of additional specific sub-goals are also put forward:

- Implement a prototype application that runs on both Android and iOS devices.
- The prototype application shall be able to interact and communicate with Advagym's IoT devices and APIs (no simulation).
- The prototype application shall also be able to interact with Sony's R&D Center Lund Laboratory CTS (no simulation).
- Internal testing and debugging of Advagym and Sony's R&D Center Lund Laboratory's systems shall be possible.
- Data from the testing and debugging shall be exportable and easily parsed.

1.3 Previous work

There are two thesis works that have been conducted at Advagym. They both handle and research different aspects of the Advagym solution for digitization of the gym:

"Interaction with IoT data to help users train smarter"

This thesis work - by Jakob Håkansson - deals with and explores new ways of utilizing the IoT data present in Advagym's systems and devices (see Chapter 2) [5].

Through a human centered design process and multiple iterations of prototypes - real-time data from IoT units are collected and presented using velocity based training (VBT) methods. The purpose of the data collection and presentation was to help users follow a predefined velocity target for a specific exercise movement.

The prototypes were implemented and developed as smartphone applications running on the iOS operating system, written in Swift. The prototypes listened to Advagym's IoT devices' broadcasts and parsed the associated data packages sent out in order to provide different user interfaces that were subject to usability tests. The usability tests determined the performance differences found in the prototypes and user interfaces - along with their understandability.

"Designing and evaluating a free weight training application"

Dominika Motylinska and Joakim Rudberg examined ways of implementing a free weight training feature and interface for Advagym's smartphone application [6]. Through a human centered process, they iterated upon and created several prototypes that resulted in three high-fidelity (Hi-Fi) prototypes running on the Android operating system.

The three apps were structured in different ways in regards to the amount of interaction necessary by the user and the amount of control and accuracy that they provided when tracking and logging. A concluding usability test with 18 test participants examined how fitness data is valuable and what level of interaction provides the best user experience.

1.4 2030 Agenda for Sustainable Development

Figure 1.1: The 17 Sustainable Development Goals detailed in the 2030 Agenda for Sustainable Development.

In 2015, all United Nations Member States adopted the *"2030 Agenda for Sustainable Development"* - a plan of action that provides a way forward for "peace and prosperity for people and the planet, now and into the future" [7]. The 17 Sustainable Development Goals (SDGs) deal with ending poverty and other deprivations - that also must go hand-in-hand with actions that aid health and education improvements, inequality reduction and spurring economic growth - all while taking on climate change and preserving our oceans and forests (see Figure 1.1).

This thesis work will try to aid in the third goal of the 2030 Agenda: "Ensure healthy lives and promote well-being for all at all ages". This will be done through the research, discussions and conclusions carried out - that will be used to better encourage and support people who are interested in physical exercise and activity.

Chapter 2

Technical background

In order to provide a more-or-less "complete" digitization solution of the gym-experience - one would distill it down into the following required parts:

- **Hardware**: in order to somehow translate the physical world into the digital, you would need devices and sensors placed and mounted in different ways throughout a gym.
- **Software platform**: interpreting/parsing the data recorded by the hardware requires different types of software to be written. Transmitting the processed data to where it is needed along with handling a large amount of users, requires some form of a flexible software platform to build various features/solutions on top of.
- **Client/application**: in order for a user to interact with the software platform and hardware, a client/an application of some form is needed; such as a smartphone application.

Each of the three parts mentioned above will in turn require specific technologies and theory in order to function. Since the project will be interacting with all of these parts in some way or another for the purpose of answering the research questions and completing the project sub-goals - it is important to understand the background of the technologies and theory necessary for doing so.

The required technologies and theory are mostly centered around computer software, computer science fields of research - along with devices and hardware containing sensors of different kinds. Advagym currently employs a lot of this in their current solution for the digitization of the gym-experience. The following sections will present these technologies and theory along with an in-depth account of Sony's R&D Center Lund Laboratory and Advagym's current solutions and related works.

2.1 Internet of Things

Internet of Things is a broad term with a broad set of definitions [4]. One way of describing it would be to define it as a network of physical objects ("things") that can be embedded with sensors, a system-on-a-chip (SoC), software and other technologies for the purpose of connecting and exchanging data with other devices and systems - over the internet.

These devices can range from ordinary household objects to intricate and sophisticated industrial tools [8]. Common examples of IoT devices would be: smart light bulbs, smart power-switches, smart ovens and other kitchen appliances. There are more than 10 billion connected IoT devices in 2022 - and experts are expecting this number to grow to 14 billion by 2024 and 25 billion by 2030 [9].

Therefore, it is only natural for a lot of companies - Sony included - who already produce and develop devices and products that are "compatible with IoT" - to be greatly interested and invested in this growing field. Another key feature that makes IoT devices an even more competitive new technology - is the possibility of implementing over-the-air (OTA) updates, meaning continuous support is possible with bug fixes and new features [10]. This saves on costs for replacement and potential recalls - along with complying with regulatory security and privacy laws.

2.1.1 MQTT

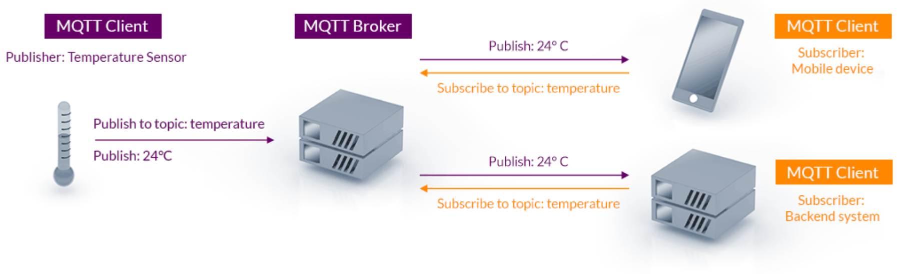


Figure 2.1: An overview of the MQTT "publish/subscribe" architecture [11].

The way in which some IoT devices communicate is through the use of MQTT (an acronym for Message Queuing Telemetry Transport). MQTT is an OASIS (Organization for the Advancement of Structured Information) standard messaging protocol that is designed as an extremely lightweight publish/subscribe messaging transport for connecting remote devices [12].

MQTT functions by having clients (devices) "publish" messages on a specific "topic". The messages are sent to a central "broker" (basically a server) - which in turn forwards the messages to the devices that are "subscribed" to the topic associated with the message [12]. For instance - a smart thermometer publishes the message "24°C" on the topic "temperature", to the broker. The broker then forwards the "24°C"-message to all devices that are subscribed to the "temperature"-topic (see Figure 2.1).

The messaging protocol has a small code footprint, uses minimal network bandwidth and due to the above-mentioned publish/subscribe system allows for messaging between device to cloud and cloud to device. This makes it uniquely apt at broadcasting messages to groups of things, making it ideal for IoT [13].

2.2 Machine learning

Machine learning (ML) is a branch of artificial intelligence (AI) which uses data and algorithms in order to imitate the way that humans learn [14]. The algorithms that drive ML are critical to its success. ML algorithms build a mathematical model based on sample data (known as "training data"), which in turn makes predictions or decisions without being explicitly programmed to do so.

Therefore, ML provides the foundation for AI systems that automate processes and solves data-based problems autonomously. There are several approaches in regards to learning and training algorithms that are used in ML [15]:

- **Supervised Learning**: training data is provided to ML systems with a directory of inputs and associated desired outputs. Through this approach, machine learning systems can understand the intended results of a given set of actions - and build optimal strategies for getting those results.
- **Unsupervised Learning**: this approach uses unstructured data sets without associated, ideal outputs. It is up to the machine learning systems to understand patterns in the data sets in order to develop strategies for behavior.
- **Reinforcement Learning**: this is typically used to train independent machine "agents" in a given system. The approach employs trial and error to come up with a solution to the problem. To get the machine to do what is wanted - the machine "agent" gets either rewards or penalties for the actions it performs. Its goal is to maximize the total reward.
- **Deep Learning and Neural Networks**: traditionally, ML and AI systems use linear or iterative approaches to machine learning. In the 1980s - researchers developed "neural network" brains with node-cluster structures and weighted decision-making strategies. This enables machine learning systems to break down complex problems into simpler ones - thereby letting the results of more straightforward problems come together as more comprehensive solutions for larger ones.

The overarching goal of ML is to replace or augment certain human capabilities. Common machine learning applications found in the real world include chat bots, self-driving cars and object detection and classification - the latter two being an important part of this thesis work [16].

2.3 Computer vision

Computer vision is a field of study/process of understanding digital images and videos, using computers. It is regarded as a subset/application of machine learning and artificial intelligence with the goal of automating tasks that human vision can achieve. This involves ways of gathering, processing, analyzing/parsing and extraction of data from the real world in order to produce relevant information [17]. The relevant information can then be used to understand the content and context of digital images and videos. Understanding the content and context can in turn provide useful and meaningful insights in order to solve various problems.

It is a multidisciplinary field of study that could be assigned as a sub-field of artificial intelligence and machine learning - that can employ the use of specialized methods and make use of learning algorithms. Computer vision also has sub-fields/domains such as object recognition, object tracking, motion estimation - which has resulted in applications throughout the field of medicine, navigation, industry and more.

Essentially, computer vision works by using a camera/multiple cameras to take pictures or videos - which is then analyzed by a (computational) device. An example of an application could be that a computer vision system can check to see if there is any food left inside of a refrigerator.

In recent years - there has been a widespread adoption of a number of tools that make such technologies more accessible and more powerful for companies/developers big and small through various pieces of software and research papers. Such tools include the open source software library OpenCV that was built to provide a common infrastructure for computer vision applications - and to accelerate the use of machine perception in commercial products [18].

The advancement seen in this field of study is yet another step leading to the real world being digitized in meaningful ways. Everything from tracking subjects with a camera, to identification of cancerous tissue/cells; these technologies will keep having enormous impact in the years to come [19].

2.4 Camera-based Tracking System

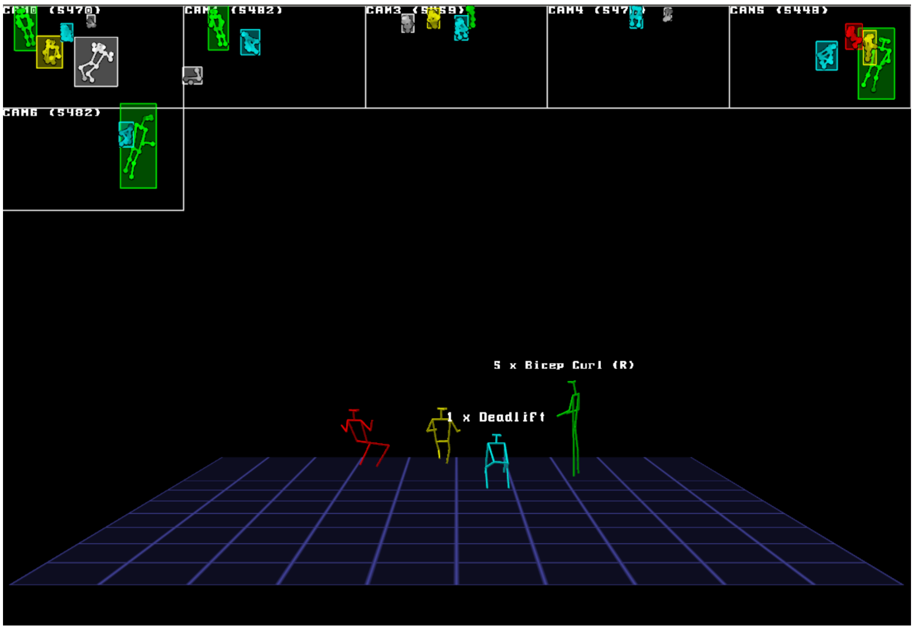


Figure 2.2: A visualization of what the CTS "sees", image courtesy of Sony's R&D Center Lund Laboratory.

Through the use of both machine learning and computer vision - Sony's R&D Center Lund Laboratory has developed a Camera-based Tracking System (CTS, a non-official term coined for the sake of this thesis work) that enables 3D object-tracking and that is trained to detect specific movement patterns.

In order to accurately track multiple objects within a space - the CTS employs a multi-camera solution. This is done in order to minimize the issue of occlusion present in many single-camera solutions for object tracking [20]. Occlusion occurs if an object you are tracking is hidden (occluded) by another object/element of the environment. Multiple cameras alleviates this problem by having partially overlapping fields of view - letting the system "view" objects from different angles where occlusions might not be present [21]. Having multiple cameras with temporally consistent video feeds, also allows matching of the two dimensional (2D) inputs - which in turn makes it possible to achieve fast individual 3D pose estimations of all humans present in the view of the cameras [22].

With the use of specific machine learning approaches - Sony's R&D Center Lund Laboratory is able to teach their CTS to classify, detect and track specific exercises being carried out by humans, in real-time. The CTS functions by constructing something resembling 3D-skeletons out of detected human subjects and their coordinates in 3D-space - that is then mapped to the physical space of the tracking area (see Figure 2.2). It is then possible to observe the tracked humans (skeletons) moving around in the tracking area - along with what exercises they perform, through the use of visualization software. The exercises that the system is currently able to detect and track includes *Bench Press*, *Bicep Curls*, *Deadlifts*, *Situps* and *Squats*.

2.5 Cross-platform frameworks

One of the goals of the thesis work was to provide Advagym and Sony's R&D Center Lund Laboratory with a smartphone application that runs on both Android and iOS devices. This would in effect mean that you would have to contend with two separate code bases - written in two different programming languages. Naturally, this also implies that every single feature of the application would need to be implemented twice - essentially doubling the workload.

Since only one person is conducting the thesis work - that would most likely result in a far-too-large workload with regards to the scope of a master's thesis work (which has a 20-week duration).

There are, however cross-platform frameworks available that offer a "solution" to this problem, such as React Native [23] and Flutter [24]. These frameworks offer cross-platform support when building smartphone applications - which affords a number of advantages over a platform-specific development/implementation. The advantages differ somewhat amongst the different frameworks available due to their specific features - but what the they all have in common is the following:

- **Code reusability**: only a *single* code base - written in a single language - is needed for the development of a cross-platform application. This heavily decreases the time, effort and cost of development and maintenance.
- **Community-driven & open source**: there is a large incentive for developers around the world to share and discuss solutions to problems on forums across the internet due to the open source nature of the frameworks. This, along with actively maintained community-built components and implementations can aid in reducing overall development time and cost [25].

An important differentiator between React Native and Flutter is found in what programming language they use. React Native uses the JavaScript programming language, whilst Flutter uses the Dart programming language. They also differ in their approach to features and components that are native to one platform - but not available on another.

React Native aims to bridge this gap by letting components and features be true to their native counterpart - whilst trying to maintain functional parity between the platforms. Flutter on the other hand does not allow components and features to differ between platforms - instead falling back to custom equivalents, thereby maintaining functional- and look-and-feel parity.

2.6 Advagym

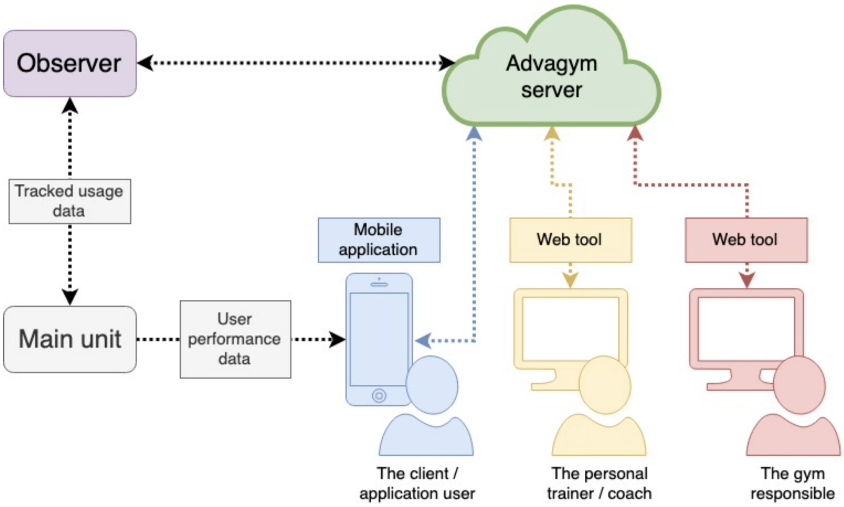


Figure 2.3: Diagram of Advagym's system and data flow to different users, courtesy of Advagym [5].

Advagym is currently looking to digitize the gym experience in meaningful ways for their users. This has required synergy between the three required parts of such a solution: hardware, software platform and client(s)/application(s).

The current system is complex - and with entirely different user groups with differing needs and requirements (see Figure 2.3). The user groups can be simplified into three different types:

- **The client/application user**: this is the "baseline" user that exercises at a gym and uses its equipment and facilities. This user interacts with the system through the use of a smartphone application on either iOS or Android. They can log their workouts and exercises automatically - if they use gym machines. Free weight exercises however, need to be manually added by the user.
- **The personal trainer (PT)/coach**: this user is able to create personalized exercises and programs that they can send to their gym-going clients (i.e. the user group mentioned previously). The PT is also able to monitor their clients' performance and progress - meaning that they have access to specific user's data.
- **The gym responsible**: they can get relevant data and information in regards to the gym's overall utilization and activity in different parts of the gym, or specific gym machines. They can also offer more general - but still uniquely specialized - exercises and programs to their members, in order to drive retention and other behavior.

In order to gather and distribute data collected within the gym - Advagym uses IoT devices that are installed at appropriate locations therein.

2.6.1 Hardware solution

Advagym's system employs three separate devices: the *main unit*, *puck* and *observer* [5]. They mimic the top-level hierarchy found in the majority of IoT solutions: device, gateway and cloud [10] - where the cloud part of the hierarchy is fulfilled by a separate Advagym server that the previously mentioned devices connect to.

To better grasp and understand the Advagym system and solution - the three devices (main unit, puck and observer) and how they interface with one-another will be explained below:

Main unit

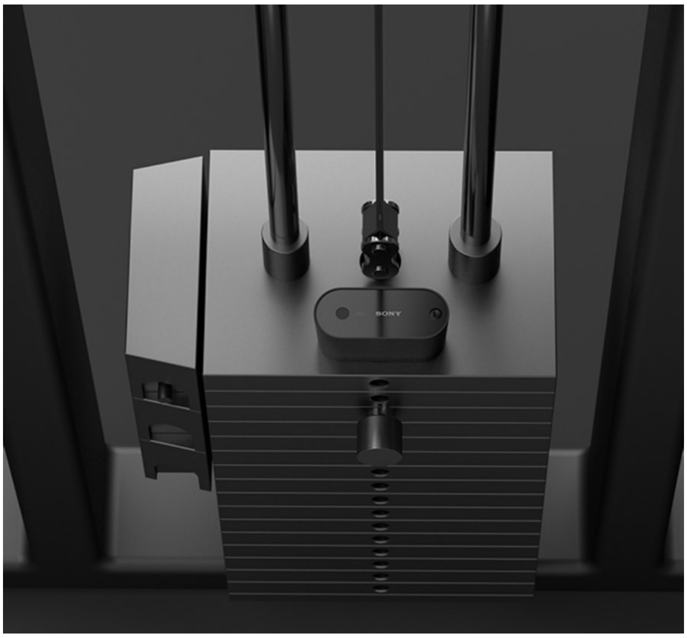


Figure 2.4: A render of Advagym's "main unit" IoT device on a gym machine weight stack, image courtesy of Advagym.

The main unit is an IoT device that can track and log performed repetitions of exercises done with a gym machine with the help of embedded sensors. Main units are installed on top of a gym machine's weight stack - which means that movements will only be along the vertical axis (see Figure 2.4). Each main unit has a unique media access control address (MAC address) that is registered in the Advagym server.

When a weight stack is moved - an accelerometer sensor wakes the main unit from "sleep mode" (a low-power state of the device) - and starts tracking movement. Movement is measured through the use of a time-of-flight (ToF) sensor that casts a laser which reflects on a piece of reflective material installed above the main unit. The measurements are then fed into an algorithm that constructs data packages for each repetition and the associated movement measured.

The data package is then broadcast multiple times through the use of Bluetooth Low Energy (BLE) - a short-range wireless technology that is used for data transmission between fixed and mobile devices over short distances. The packages are received by both the user's Advagym smartphone application to interpret - as well as the observer, to log.

Puck

Figure 2.5: A render of Advagym's "puck" IoT device, image courtesy of Advagym.

In order for a user to log repetitions and exercises performed on a certain gym machine automatically, a connection between the user's Advagym smartphone application and the main unit needs to be established. This is the primary function and purpose of the puck (see Figure 2.5). This is done through the use of two different technologies - depending on what platform and what smartphone features that the user has:

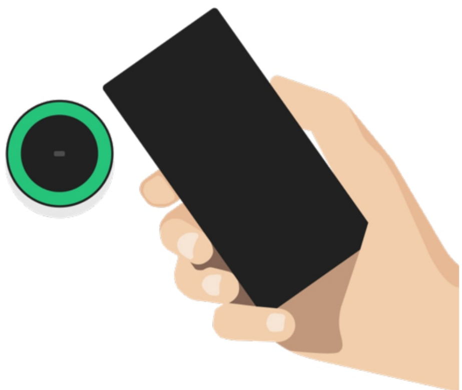


Figure 2.6: Interaction showing a user "tapping" an Advagym puck, image courtesy of Advagym.

For users with a smartphone running iOS - a BLE data package is received from the puck when the user "taps" it with their smartphone (see Figure 2.6). "Tapping" a puck equates to a user physically touching the puck with their smartphone instead of their hand. When a puck is covered by the smartphone - a proximity sensor triggers, which in turn activates the BLE data package sending functionality.

For users with a smartphone running Android that has near field communication (NFC) capabilities - the same interaction detailed above is performed. However, data packages are sent over NFC instead of BLE. Some smartphones running Android do not have NFC capabilities; in that case, the puck behaves precisely as with iOS smartphones (i.e. BLE used).

The puck itself is positioned where it is clearly seen on the gym machine itself, or in close proximity to the gym machine where it is obvious-enough for the user to understand that the puck is associated with that gym machine.

Observer

Figure 2.7: A render of Advagym's "observer" IoT device, image courtesy of Advagym.

The last piece of hardware used by Advagym is the observer. Its main purpose is to act as a link between the main units and the Advagym server. It is placed in the same area as the main units and gym machines. It constantly listens to all of the broadcasts performed by the main units - thereby keeping track of the status of all the connected gym machines (important for tracking gym utilization and building statistics). The data packages received in the broadcasts are then sent to the Advagym server over the internet - making the observer act as the "gateway" of the connected IoT devices (i.e. main units and pucks) [10].

2.6.2 Software solution

Advagym has - as mentioned above - three different user groups: the client/application user, the personal trainer (PT)/coach and the gym responsible. In order to support these user groups, Advagym's software solution is comprised of three parts: the server (& software platform), the smartphone application, and the web tool - that are detailed below.

Server & software platform

All of the data collected from the main unit devices in the gym - is sent to Advagym's server with the help of a close proximity observer device. The server differentiates the different main units through their unique MAC addresses. Each main unit is then registered by the Advagym server through its MAC address, along with its location, gym machine (exercise) and also the MAC address of the associated puck used to wake and interact with it. The main unit and its puck are therefore paired together and are registered as a single unit by the server.

Data collected from the main units is made available to the other parts of Advagym's software solution through their software platform. The software platform is built with specific application programming interfaces (APIs) that are accessed by the smartphone application and the web tool - allowing them access to certain stored data on the server.

As a security and privacy measure - certain APIs and data are only available to users with a certain account type (i.e. "admin", "PT", etc.). For instance the non-anonymous data generated by users of the smartphone application is owned by those users. Such data involves inputs from the user regarding the tracked and logged data. However, non-anonymous user data can be shared to and accessed by a PT (with the associated account type) *if* the user in question explicitly allows for it [5]. Anonymous data - such as repetitions performed on a gym machine - is relevant to the gym responsible and is therefore shared with that account type.

Smartphone application

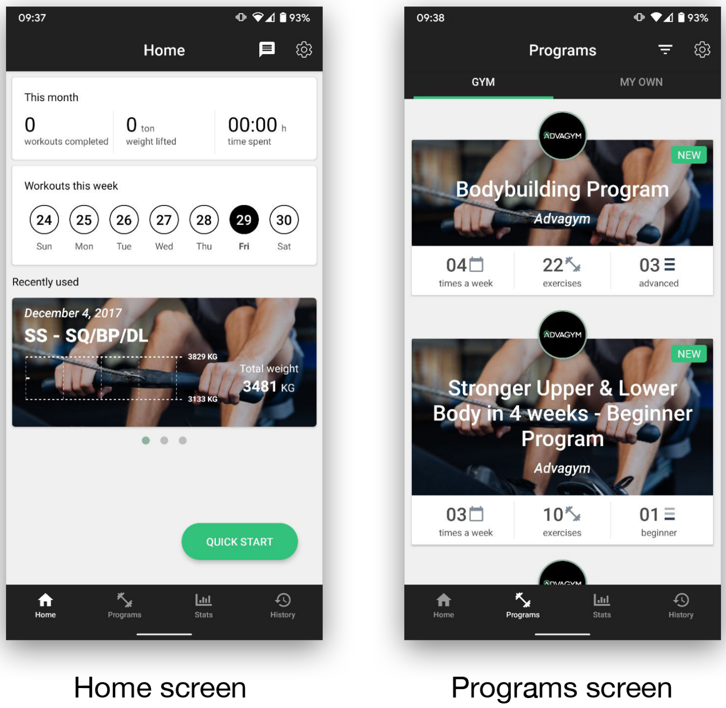


Figure 2.8: Screenshots of two screens present in the Advagym smartphone application (running on the Android operating system).

The client/application user group uses Advagym's smartphone application in order to interact with their hardware (devices) and system (see Figure 2.8). The user group encompasses the common "gym-goer"; i.e. the people who exercise and workout in the gym. The application guides the user through the gym-going experience and Advagym's features and solutions. The application itself has several specific features:

- **Logging**: automatic or manual logging (for free weight exercises or custom exercises) of performed repetitions (reps) and sets during each workout session.
- **Programming**: users can access already-made workout programs that contain a structure of exercises with associated sets and reps.

- **Insight**: all of the performance data tracked and logged by the app and the user is summarized and displayed to the user with relevant and intuitive statistics.
- **Training support**: instructive and educational videos for performing exercises videos are available - along with rest timers, training program descriptions and much more.

Web tool

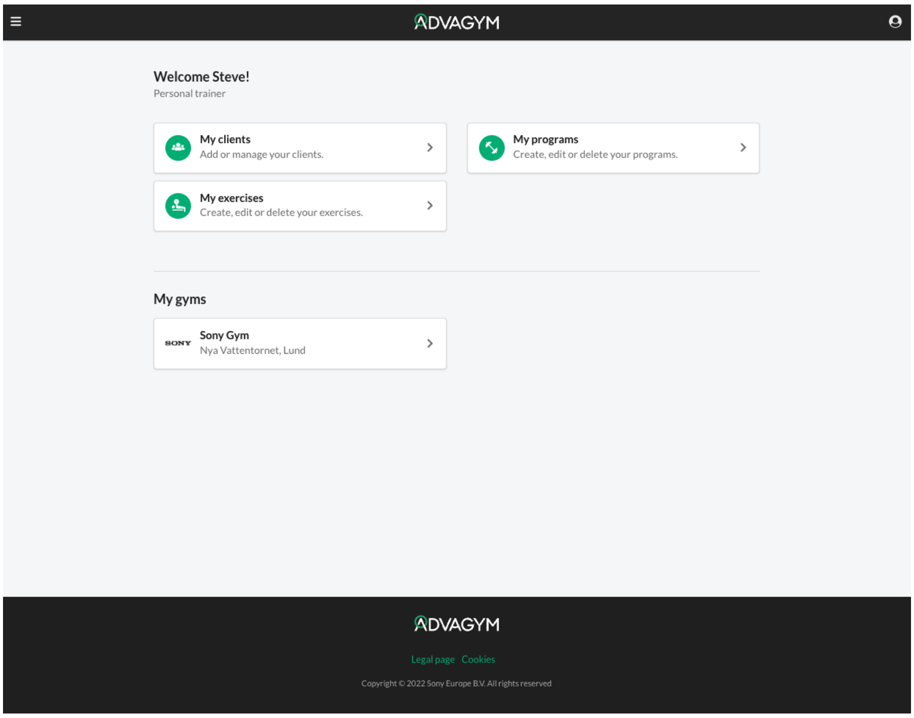


Figure 2.9: A screenshot of Advagym's web tool, image courtesy of Advagym.

The PT and gym responsible user groups both interact with Advagym's solution through the use of the web tool (see Figure 2.9). The web tool mainly serves as an overview and offers management of the gym, its users and Advagym's solutions therein.

For the PT, the main features of the web tool are: client & group coaching/management as well as personalized content distribution (e.g exercises, programs etc).

For the gym responsible, the main features of the web tool include: gym content distribution, member and staff management, hardware configuration and status and gym statistics (e.g utilization, retention, content usage etc).

2.7 Related work

There are currently a lot of studies and research being done in the field of computer vision & machine learning - and the applications thereof - that are relevant to this thesis work. The results of this have slowly begun to make its way into commercial products available to the common consumer.

Fitness and gym-oriented applications are a mature ecosystem with user interface patterns relating to their intended use becoming more standardized and widely adopted.

Regarding the issue of evaluating how potential users would prefer to track and log their free weight exercises - a separate thesis work has also been conducted at Advagym (see Section 1.3).

2.7.1 Computer vision and machine learning

Amazon Go Stores

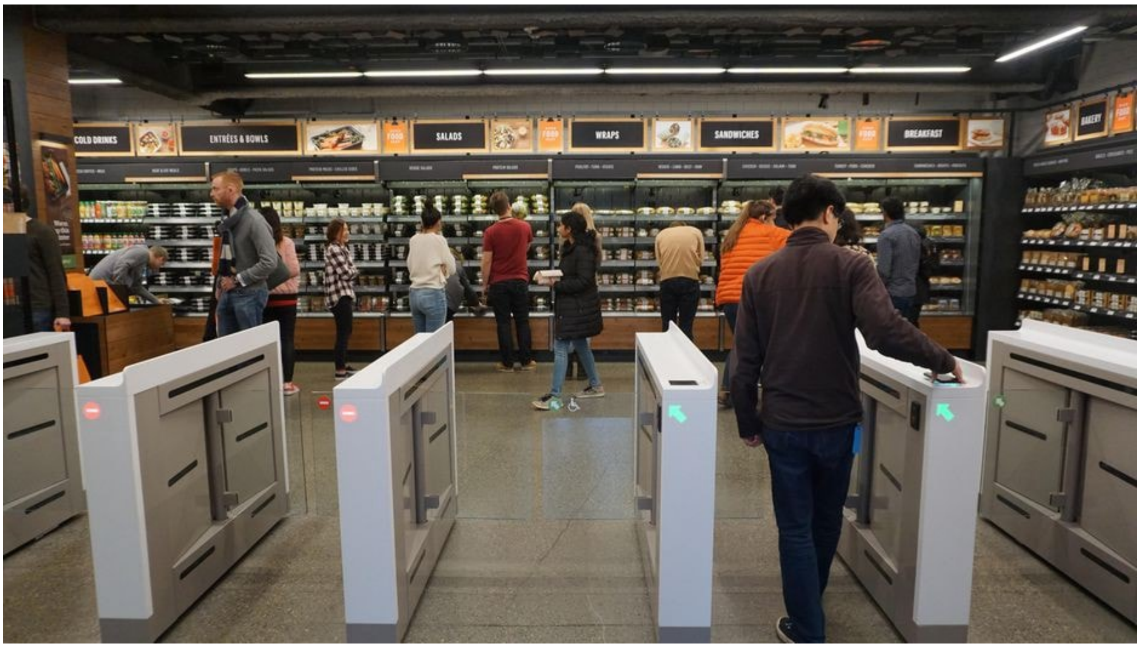


Figure 2.10: A customer entering the Amazon Go store by scanning their smartphone's screen in the turnstile.

Amazon is primarily known as an internet marketplace for goods & products, where users browse their website (or use their apps) in order to make purchases that get shipped to their doorstep.

In 2018, Amazon launched a new physical shopping concept: Amazon Go. Traditionally when shopping at a physical location - one would need to pass through some form of a check-out process whereby each individual item that the customer wants to buy - needs to be entered into a system when leaving the store. The items are then paid for using a physical credit card or cash - usually with the help of a human cashier [26].

When customers visit an Amazon Go-store - they first enter through a turnstile where their smartphone's screen (with the Amazon smartphone application opened) is scanned (see Figure 2.10). Then - they simply walk around the store and put food items into their shopping cart. When the customer has put everything they need into their shopping cart - they leave the store and the total amount for the purchases, is automatically charged through the credit card registered with their Amazon account. No further interaction - except the initial scan in the turnstile - is required of the customer.

All of this is done through the use of multiple cameras utilizing computer vision and deep learning algorithms together with "sensor fusion" (i.e. different sensors such as weight- and proximity sensors working in concert) [27]. Users and individual food items are tracked with object detection and classification techniques. The precise nature of the system and techniques used is not disclosed by Amazon, at the time or writing

GymCam

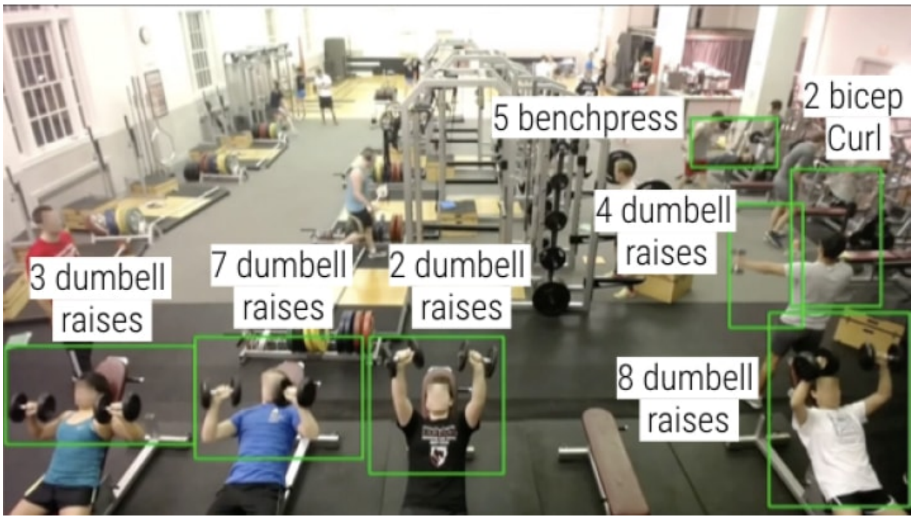


Figure 2.11: GymCam's tracking and exercise detection solution, with green boxes around tracked objects and text with the tracked exercise and repetition count.

GymCam is a proposed solution that detects, recognizes and tracks exercises in a gym through means of computer vision and machine learning algorithms. The system utilizes a single camera solution to track motion and assumes that most repetitive movements over time in a gym are to be regarded as "exercises in progress" - thereby differentiating random movements from exercises [28].

The paper outlining GymCam - released in 2018 - also claims that the proposed solution is able to track hundreds of users simultaneously - whilst still being able to detect and track the individual exercises and the corresponding repetitions being carried out by the users (see Figure 2.11). The system can also handle occlusion due to its focus on detection of repetitive movements in favor of trying to accurately estimate body keypoints (i.e. skeletons) as with typical multiple camera solutions.

2.7.2 Gym fitness applications

With a cursory glance at some of the top gym fitness applications (such as Strong [29], Jefit [30] and FitNotes [31]) on the largest application marketplaces (the AppStore and Google Play Store) - one can see that the interface that gym-goers encounter when logging their repetitions and sets of their exercises are quite similar in terms of layout. Due to the fact that the fitness application market today is mature and fast growing - it is only natural for a consensus of sorts forming around what user interface practices and patterns are preferred and/or expected by users [2, 32].

As discussed previously in the background-section of this report - going to a gym, tracking and logging your exercises and workouts is a complex task. Couple that with programs and new training methodologies and exercises steadily making their way into the gym-going community - gym fitness applications need to find user interface and user experience patterns that support and help the user where necessary and build upon those [33].

Chapter 3

Methodology

During the thesis work and when developing the prototypes, different methodologies were used. The idea was to follow established processes and principles when exploring interaction- and user experience design of systems and solutions utilizing the relatively new fields and technologies of computer vision and IoT.

The following sections will explain the methodologies, principles and processes used in detail:

3.1 Fundamental principles of design

When developing and implementing a product or solution - one must make sure that all aspects that the users interact with are easily accessible, simple and intuitive for the user. Norman has specified seven fundamental design principles that should be taken into account when working within a design process [34].

The seven design principles are specifically used to gauge the current ease-of-use and intuitiveness of the product or solution in question:

- **Discoverability**: the user is able to determine what actions are possible along with the current state of the product/solution.
- **Feedback**: some way of letting the user know that the system is working on their request or having registered their action.
- **Conceptual model**: the design presents all the information needed for the user to have a sense of understanding and a feeling of control in regards to the product.

- **Affordance**: is the perceived action and actual properties of an object that help the user determine its operation. For instance, a button that you use to call an elevator affords you to push it and a chair affords you to sit on it.
- **Signifiers**: refers to any mark, sound, or any other perceivable indicator that communicates appropriate behavior to a person. It helps a user understand the design or feedback being presented to them.
- **Mapping**: is the relationship between two sets of things. Basically, which control is connected to what action present in the product/solution. This is enhanced through spatial layout and correlation, for instance.
- **Constraints**: physical, logical, semantic and cultural constraints restrict the kind of interactions that can take place and therefore help reduce the amount of information a user needs to process.

3.2 Iterative design approach

Iterative design is an approach that developers, designers and others use to continually improve and build upon a concept, design, product, etc..

Generally, one would start by creating a prototype and then proceed to test it. After having tested it - one would alter and tweak (*iterate*) and perform additional testing on the now-revised prototype. This *cycle* will then be repeated until the concept and solution converges, since you progress towards your desired result each time you iterate the product [35].

The iterative design approach is widespread across many industries, including software development (which is highly relevant to this thesis work). Most so-called "Agile" software projects use an iterative design approach - incrementally improving the product with each cycle or "sprint" [36]. The result/end of one iteration then becomes the starting point for the next round of development.

You can often find the iterative design approach being applied within larger, more comprehensive design processes, such as the human-centered design process.

3.3 Human-centered Design Process

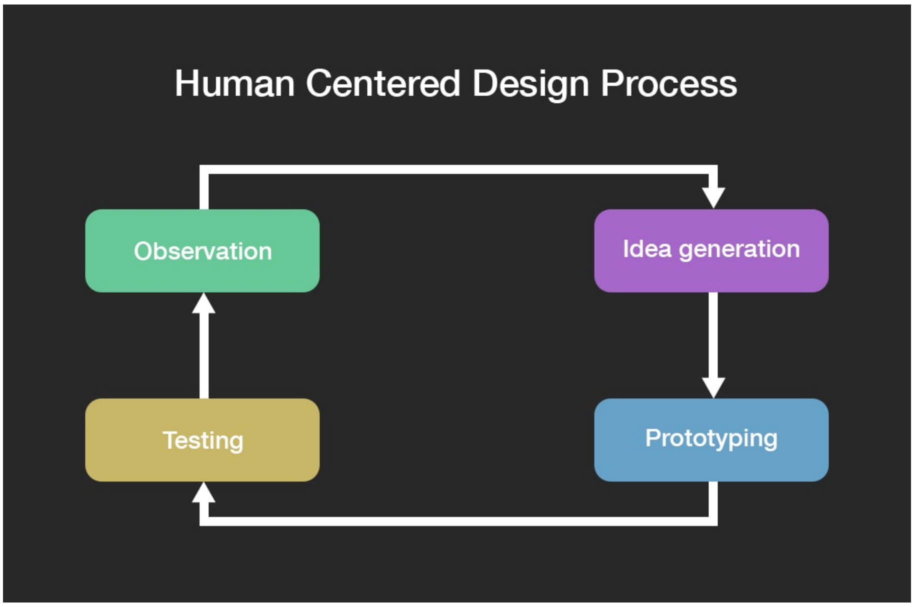


Figure 3.1: An overview of the different phases of the iterative human-centered design process.

The human-centered design (HCD) process is a way of going about designing products or solutions that puts people (humans) first. When abiding by the HCD process, product creators attempt to find solutions to problems by involving the "human perspective" in all steps and facets of the problem solving process. HCD is about understanding and catering to the people who use the products - what their needs, wants, behaviors, emotions and tasks are.

Cognitive science and usability engineering expert Don Norman summarizes the four main principles of HCD, thus [37]:

- **Be people-centered**: focus on the people you are trying to cater to and their context, in order to create things that are appropriate for them.
- **Understand and solve the core problems**: if you do not solve the root causes, the underlying fundamental issues - the symptoms will just keep returning.
- **Everything is a system**: visualize everything as a system of interconnected parts - complications/errors most often result from the inter-dependencies.
- **Continually test and refine**: continually iterate to make sure that your small tweaks truly meet the needs of the people you focus on.

The HCD process also implies that you take an iterative design approach (detailed in the previous section); meaning that throughout the HCD, you go through multiple cycles of the same exact process (see Figure 3.1). Norman details four specific phases [34]: *Observation*, *Idea generation*, *Prototyping* and *Testing*. These are discussed in the following sections:

3.3.1 Observation

The first phase in the HCD process/cycle is observation. The most useful observations take place in the real world - not in a controlled setting like a lab.

One way of going about this is through applied *ethnography* - which equates to observing users in their "natural" environments, carrying out the activity/activities of interest [38]. This will give the most comprehensive picture of a user's need, how their activities/tasks are structured and what their expectations are in regards to the result/outcome of said activity/task. It's also important to note that one needs to make sure that the people being observed are part of the intended audience for the final product (i.e. the "end-users").

When conducting observation - it's important to distinguish activities from tasks. Take driving a car for instance. "Driving" is the activity - but "steering" and "checking mirrors" are tasks that serve the overarching activity. Due to the fact that each of the tasks serving the activity are identified - one can break down the problem of driving into smaller parts, thereby making learning to drive easier [34]. The activity helps with making sense of the disconnected components (tasks).

3.3.2 Idea generation

The second phase in the HCD process/cycle is idea generation. From the observations conducted - the necessary background knowledge is gained in order to both discover and define the problem(s) that needs to be solved. The act of exploring solutions for the discovered problem(s) equates to idea generation [34]. During idea generation, it is helpful to practice and remember the following "guidelines":

- **Focus on quantity over quality**: generating as many different ideas as possible will increase the odds that at least one of them will be useful in some context.
- **Do not "censor" yourself**: do not dismiss ideas before they have a chance to be developed or expanded upon. Even if an idea is not considered relevant or helpful in the end - it can still spark useful discussion and reflection.
- **Question everything**: posing "obvious" questions allows for the opportunity to re-examine basic, fundamental ideas. Oftentimes, that very concept is what leads to the most innovative ideas and designs.

There are many different types of techniques and methods that can be applied when conducting idea generation:

Brain- & bodystorming

Brainstorming is a heuristic method of creative problem solving where the purpose is to generate as many ideas or solutions to a problem or a group of problems as possible - in one intensive session with often multiple participants [39]. The way in which the session is carried out can vary, everything from writing on sticky-notes, drawing and sketching on pieces of paper to having moderated discussions.

During the session, participants are encouraged to freely share and submit their ideas and exchange their views and opinions. It is however important to remember the guidelines stated above: postpone any criticism and encourage building upon ideas instead of discarding them. One way of further stimulating idea generation in a session is to employ a technique called *SCAMPER* (Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, Reverse) [40].

Bodystorming can be seen as an extension of brainstorming. However, instead of just verbally and theoretically putting oneself in other people's shoes - one would instead **physically** experience what people are going through in terms of the task at hand or the problem needing to be solved. In practice, this means that participants of the bodystorming session will adopt the spatial, cognitive, temporal, emotional and motivational perspective of the "potential user".

Workshops

Workshops are defined as a hands-on session where small groups of professionals (and/or non-professionals) work together in a creative manner [39]. The fact that you work together in a group-setting, allows for interaction between participants that trigger idea generation. The participants do not need to have an insight or understanding of the topic of the workshop. Generally, workshops begin with an introduction of the topic at hand so that all group members are all on equal footing.

During workshop sessions - one can examine concepts, technologies and/or practical examples pertaining to the overarching problem/solutions that you're looking to solve. The hands-on part of the session relates to giving the participants a chance to physically test and work with existing concepts or technologies. In general, a workshop consists of different activities, ranging from free-form discussions to practical implementations of simple prototypes.

3.3.3 Prototyping

The third phase in the HCD process/cycle is to explore the most promising or relevant ideas or solutions in detail. A prototype should be seen as an expression of a specific design intent. Prototyping allows for presentation of the aforementioned promising or relevant ideas, and then being able to study and analyze the ideas and solutions being interacted with by potential end-users. In the context of software development, a prototype is a simulation of the final interaction between a user and interface [41].

Prototypes do not necessarily reflect the final products/solutions; they often have various degrees of *fidelity*. The fidelity of a prototype refers to how well - or how closely - it conveys the look-and-feel or function/feature parity of the final product/solution. More specifically, fidelity can vary in areas of:

- Visual design & layout
- Content
- Functions/features/interactivity

Generally, the HCD process (and other design processes) start out with Lo-Fi prototyping during the first cycle(s) - whilst Hi-Fi prototyping is first seen during the last cycle(s).

Lo-Fi prototyping

As the name might suggest - Lo-Fi prototyping offers a fast, cheap and efficient way of iteration and trying out different ideas and solutions [42]. The most important role of Lo-Fi prototypes is to be able to analyze and test overall functionality and fundamental ideas - rather than the visual appearance of the product/solution.

In terms of the fidelity areas mentioned above, Lo-Fi prototyping generally has the following characteristics:

- **Visual design & layout**: the bare minimum of the final visual design and layout are presented.
- **Content**: again, only the bare minimum of key components/elements of the content are presented.
- **Functions/features/interactivity**: most if not all of the functions and features are *simulated* instead of being implemented.

A popular example of a low-fidelity prototyping technique is that of *paper prototyping*. Paper prototyping allows you to prototype interfaces without the need of using digital software tools. The technique involves creating hand drawings of different screens or views that represent the user interface of the product/solution.

Hi-Fi prototyping

Hi-Fi prototyping on the other hand - aims to appear and function as similarly as possible to the actual finished product/solution [42]. These prototypes are usually only created when a deep understanding of what the problem is and how a solution/product should be implemented has been reached. They are at the opposite ends of the spectrum when it comes to Lo-Fi prototyping offerings: it is both a time-consuming and expensive way of trying out different ideas and solutions.

Hi-Fi prototyping generally has the following characteristics:

- **Visual design & layout**: both the visual design and layout are as detailed and realistic as possible with regards to the final product/solution: all of the different components and graphics are being presented here.
- **Content**: most or all of the content is presented in a real (or simulated) fashion.
- **Functions/features/interactivity**: most or all of the functions and features of the product/solution are implemented. If a specific function/feature does not have a fully functional implementation - the feedback that user would be presented with upon interaction, would be *realistically* simulated at the very least.

Hi-Fi prototyping in the context of software development usually implies that the prototype has either been implemented through *digital prototyping tools* or *programming code*.

Digital prototyping tools such as AdobeXD [43], Figma [44] or Sketch [45] allows for creation of visually rich prototypes complete with interactive elements and complex animations. These are often faster and cheaper to produce than prototypes implemented by way of programming code. Coded prototypes however, are able to closer resemble the actual final product/solution - since they can have fully functional implementations of functionality/features. In contrast - prototypes created in digital prototyping tools, are stuck with having *simulated* functionality (in most aspects).

3.3.4 Testing

The last phase of the HCD process/cycle is testing. Testing is what directly fuels the iterative design approach used in the HCD process.

After having created prototypes of varying fidelity - members of the target group are invited to test and interact with these prototypes by way of different testing techniques. The *data* and *feedback* gathered from these tests are what make iteration possible since they are both used to inform what changes, fixes or tweaks are applied to the next iteration of the prototype currently being tested.

Testing will be performed multiple times during the HCD process. It is especially favorable to conduct testing during the early stages of the process - since larger changes, later on in the process will be more expensive to implement [34].

Different kinds of approaches, techniques and techniques of testing are more or less suited during different stages of the HCD process. Two examples of this are *exploratory testing* and *usability testing*:

Exploratory testing

One of the testing techniques suited for early stages of the process is *exploratory testing*. Exploratory testing does not have defined test cases created in advance - but instead, testers evaluate and interact with the solution/project in a more free-form manner [46]. The purpose of the test is for testers to explore the solution/project in order to identify and document potential issues and shortcomings. This leads to feedback specifically suited for idea generation in an early stage of the HCD process.

Usability testing

In usability testing - test participants are given tasks to perform whilst data is continually being collected. It is important that the collected data is relevant to the product/solution being tested. Examples of such data points are: completion time, the amount (and type) of errors and the overall completion rate of tasks. Compiling and analyzing this data can then show what aspects of the solution/project needs to be addressed for the next iteration of the prototype [47].

A common tool used for usability analysis is the System Usability Scale (SUS) questionnaire, that is usually presented to the test participant after having tested a prototype:

System Usability Scale

SUS is a standardized questionnaire for analyzing the usability of a system. It is intended to be used to evaluate a wide variety of products and services, including hardware, software, mobile devices and applications. It was originally designed to be a "quick & dirty" method - but is now considered the industry standard for usability evaluation [48].

The questionnaire is comprised of ten items (questions) with five response options for the respondents; from "strongly agree" to "strongly disagree" (a "Likert scale" where "strongly disagree" equals a 1, and "strongly agree" equals a 5) [49]. The specific questions of the questionnaire can be found in Appendix B.

Using the results of a number of SUS questionnaires for a specific usability test - a "SUS score" can be obtained. The score ranges from 0 to 100 and is calculated through the use of the following equation (Equation 3.1):

$$\textbf{SUS score} = 2.5(20 + \sum \textbf{Even numbered questions} - \sum \textbf{Odd numbered questions}) \quad (3.1)$$

In order to evaluate what the SUS-score infers about usability - one can use the curved grading scale (CGS) (see Table 3.1). The grading scale was derived from over 446 studies and 5000 individual SUS responses - in order to provide letter grades for SUS-scores [50]. A mean SUS score of 80 (letter grade $A-$) indicates an above-average user experience and usability [51].

Table 3.1: The grading scale interpretation of SUS Scores [50].

SUS Score Range	Grade	Percentile Range
84.1–100	A+	96–100
80.8–84	A	90–95
78.9–80.7	A–	85–89
77.2–78.8	B+	80–84
74.1–77.1	B	70–79
72.6–74	B–	65–69
71.1–72.5	C+	60–64
65–71	C	41–59
62.7–64.9	C–	35–40
51.7–62.6	D	15–34
0–51.7	F	0–14

3.4 Project work structure

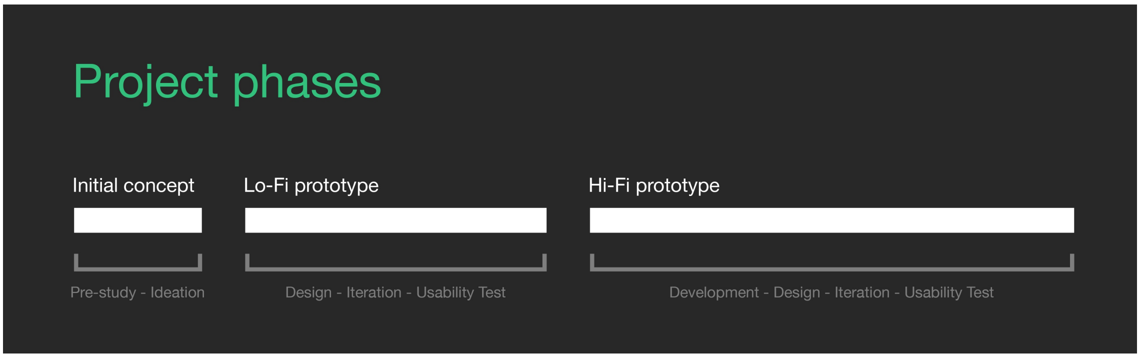


Figure 3.2: An overview of the thesis work and its three phases.

The entire thesis work was divided up into three larger parts or *phases* (see Figure 3.2):

- Initial concept
- Low fidelity (Lo-Fi) prototype
- High fidelity (Hi-Fi) prototype

The first phase involves a pre-study, ideation process and brainstorming sessions which lasted around two weeks. The results of this first phase led to the second phase where a Lo-Fi prototype was developed, which lasted around five weeks. After multiple iterations and usability testing of the Lo-Fi prototype - the last phase began: the development and implementation of a fully-functional (i.e. with no simulated parts) Hi-Fi prototype, which lasted around eight weeks.

The results and feedback from the usability tests and iterations of the Lo-Fi prototype informed the design and features of the Hi-Fi prototype.

Chapter 4

Initial concept

Figure 4.1: An overview of the current active phase of the project.

For the first phase of the project (see Figure 4.1) - the goal was to arrive at an understanding and overall concept/solution for the research questions (see Section 1.1) as well as the sub-goals (see Section 1.2).

This would encompass the first cycle of the HCD process and the corresponding activities and techniques for doing so. Detailed below are the different activities performed during this initial concept phase:

4.1 Project pitch and pre-study

Leading up to the thesis work - the proposed project was pitched, by Advagym's Lead Front-end Developer and Software Architect, Jakob Håkansson:

Advagym, together with Sony's R&D Center Lund Laboratory wanted to explore a solution for automatic free weight exercise tracking, using Sony R&D Center Lund Laboratory's CTS. They wanted a *fully functional* smartphone application (running on the iOS and Android operating system) - with *no* simulated parts. This application would then be used as a test-bed for further development and evaluation of the CTS, as well as to study the business proposition/the customer interest of such an automatic free weight exercise tracking system. Jakob Håkansson would act as the supervisor at the workplace (Advagym).

From the pitch - research questions and sub-goals for the thesis work were formulated (see Section 1.1 and Section 1.2 respectively). The questions and sub-goals were then discussed with the supervisor at Lunds Tekniska Högskola, Associated Senior Lecturer Günter Alce - together with the Advagym supervisor. From the discussion, a "Master's Thesis Scope" detailing the proposed thesis work and its planning, was written. It was approved, which meant that the thesis work could begin in earnest.

First, a comprehensive literature and pre-study was conducted - in order to get a detailed view of the current understanding of the problems associated with the technologies that were to be utilized in the thesis work (specifically computer vision, machine learning and IoT devices - detailed in Chapter 2). Members of Advagym's team of engineers were interviewed and documents were shared, detailing Advagym's current solution and hardware.

4.1.1 Exploring the Camera-based Tracking System

Figure 4.2: Picture of the gym located in the Sony offices.

A vital part of the system and solution, with the goal of automatic free weight exercise tracking - was the camera-based tracking system (CTS). It was therefore important to understand precisely how the system functioned.

Sony's R&D Center Lund Laboratory has set up a space in Sony Lund's on-site gym, to test and evaluate the CTS (see Figure 4.2). The gym has a number of different gym machines that

Advagym uses to test their hardware and software solutions. There are also a number of free weights equipment available: barbells, dumbbells, kettlebells and more.

Through a large TV-screen facing the gym area - a visualization of the CTS's tracked area is shown to the persons utilizing the gym and its equipment (see Figure 2.2). This corresponds to a "wireframe" 3D-space with simple "skeletons" representing the actual humans moving around in the space. Each skeleton is differentiated by its color. Whenever a user completes a repetition of one of the tracked exercises (see Section 2.4) - text appears above the head of the corresponding colored skeleton with the name of the exercise and the amount of reps performed.

Understanding the constraints and boundaries of the CTS was also investigated through bodystorming sessions. The cameras used within the CTS were placed along the ceiling - each covering certain portions of the gym area. Through trial and error - it was possible to determine that there were areas of the gym that were not covered - or where the tracking performance (in terms of accuracy) was severely diminished. This was generally due to the fact that the area in question was not fully covered by a single camera - or that the area was being fully occluded by some object (such as a gym machine). In those cases - the system loses track of the user and their skeleton disappears from the visualization (and then returns, when the user is "in view" again).

4.2 Brainstorming and paper prototype

After having tested and explored the CTS - multiple brainstorming sessions took place in order to develop an overall concept for the smartphone application. Tools such as a whiteboard, pen and paper were used. The following constraints and requirements of the CTS were noted:

- The system does not have 100% accuracy in terms of repetition detection.
- Users need to be within a specific area of the gym, covered by the cameras used.
- In certain use cases, the system can lose track of a user (even though they are "within" the tracked area).
- Only the exercises that the system is currently able to detect - are tracked and logged (i.e. there is no possibility of "custom" exercises unless the system is re-trained).

From these constraints, it was determined that there were three main areas that needed to be taken into account when devising the initial concept:

- **Connection**: how would a user "connect" to the tracking system?
- **User flow**: what steps would the user need to take in order to ideally interact and use the system?
- **System/user error**: what modes of feedback would need to be utilized in order for the user to be able to respond to errors?

In order to test and gain feedback on the initial concept, a paper prototype was constructed, dealing with the above-mentioned areas.

4.2.1 Connection

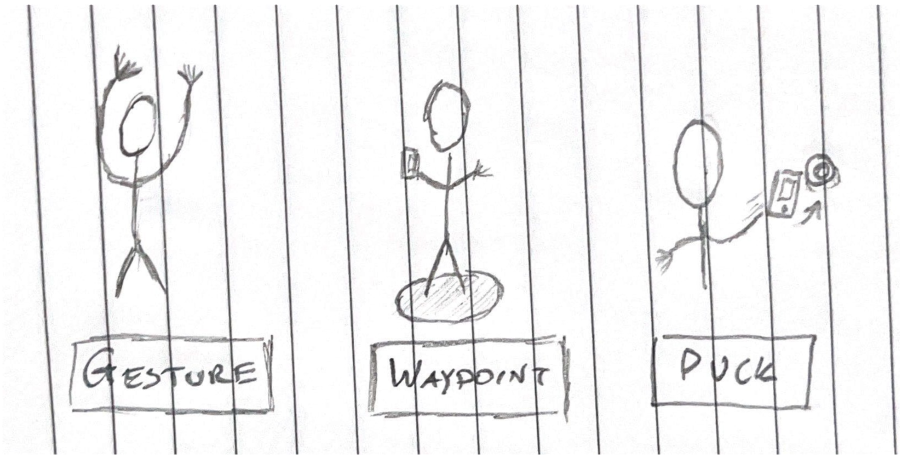


Figure 4.3: Sketch of users carrying out the three proposed connection methods.

In order for a user to connect to the system and start using it - one would somehow need to bind (link) the "skeleton" (shown in the visualization of the CTS) - with the user.

Three different methods regarding how a user would do this was proposed (see Figure 4.3):

1. **Gesture**: if the user performs a specific gesture - such as holding both arms above their head - the system can see which skeleton is doing that exact gesture and link the user.
2. **Waypoint**: if the user presses a connect button on their smartphone app, whilst standing at a precise location somewhere in the tracking area - the system would know that the skeleton at that precise location (at that very instant) - is the user.
3. **Puck**: works identically to the Waypoint-method, but the user taps a puck with their phone instead of pressing the connect button on their smartphone app.

The reasoning behind these proposals was that the CTS would not need/only require a small amount of additional implementation in regards to detection or further training. If more advanced training and additional features could be implemented, then there would've been more to consider, such as the possibility of face detection or other means of identification - of the user.

4.2.2 User flow

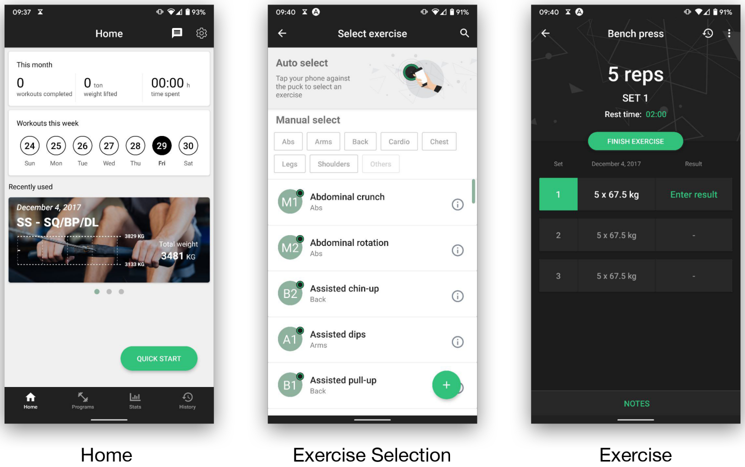


Figure 4.4: Screenshots of Advagym's current Android application's "Home", "Exercise Selection" and "Exercise" screens.

Since the resulting smartphone application of the thesis work would be used to evaluate a free weight exercise tracking solution for Advagym - it made sense to look at Advagym's current smartphone application for user flow inspiration (see Figure 4.4).

Drawing from Advagym's app - and having considered a couple of different user stories, the following "base-line" user flow was established:

1. The user first encounters some form of "Home"-screen.
2. From the "Home"-screen - the user is able to start a workout.
3. An "Exercise Selection"-screen is shown, where the user chooses what exercise he/she wishes to perform.
4. After selecting an exercise - the "Exercise"-screen is shown.
5. On the "Exercise"-screen - some form of instructions for connecting to the CTS is shown.
6. Whilst performing repetitions (i.e. being in the middle of a set), a "Current Exercise"-screen is shown.
7. When the user is done with their set, they are returned to the "Exercise"-screen.
8. When the user is done with their workout, they can return to the "Home"-screen.

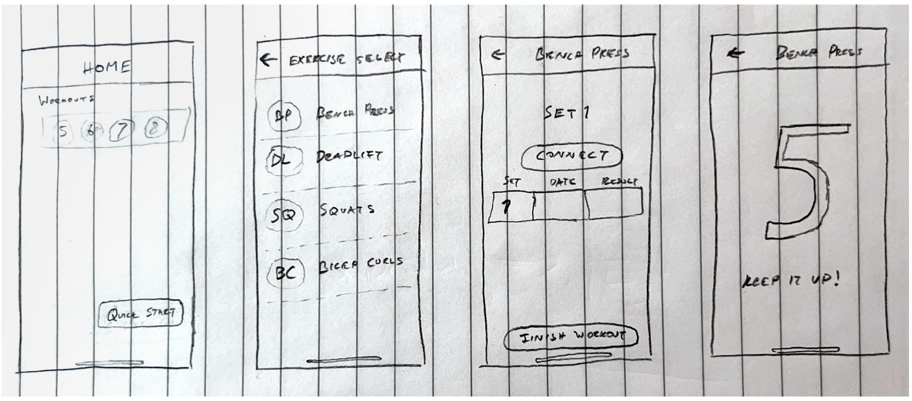


Figure 4.5: The four paper prototype screens: "Home", "Exercise Selection", "Exercise" and "Current Exercise".

The four screens ("Home", "Exercise Selection", "Exercise" and "Current Exercise") were made into paper prototypes (see Figure 4.5). The base-line user flow (mentioned above) was supported by these paper prototypes through the clicking of buttons and interacting with different elements on the screens.

4.2.3 System/user error

After having tested the CTS - it was apparent that the user would most likely encounter tracking-related issues during use. Therefore, it was decided that the amount of possible user error needed to be minimized so as to not add to the total amount of errors (which in turn could affect usability). Since the only way that the user can interact with the system is through the smartphone application - the application itself needed to have feedback affordances.

Due to the fact that users would most likely have their smartphone on their person, or close to where they are performing exercises - different types of feedback could be used, such as: haptics, visual or sound. Feedback was however not included - or added to the paper prototypes due to the focus on the base-line user flow and further ideation.

4.3 Supervisor meetings and sync

Throughout the initial concept phase - there were weekly meetings with both supervisors - along with almost daily "syncs" (short meetings) with the Advagym supervisor. The weekly meetings were focused on the overall project and the "Master's Thesis Scope"-document - whereas the daily syncs were focused on the current progress of ideation and prototyping.

The Advagym supervisor tested and reviewed the proposed paper prototypes and user flows through *exploratory testing* methods. They were deemed feasible - but needed refinement in terms of the user connection: it was not exactly understood how this would function or if it *could* function. The amount and what types of features and functions needed to be included in the resulting smartphone app (prototype), was also not fully understood.

Therefore, the next step was to set up meetings and interviews with the other stakeholders of the project (mainly the team at Sony's R&D Center Lund Laboratory) - so as to verify exactly how the CTS's underlying code/structure functioned - and what they wanted out of the project.

Chapter 5

Lo-Fi prototype

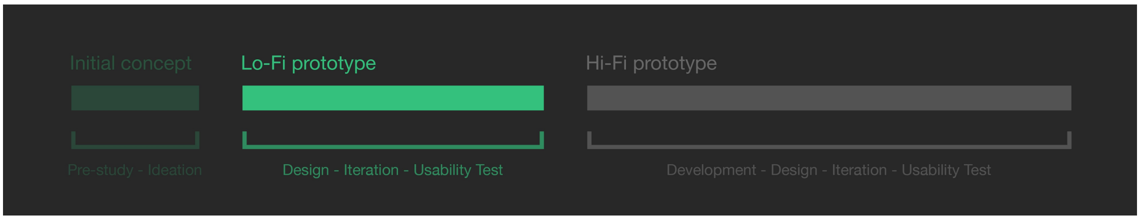


Figure 5.1: An overview of the current active phase of the project.

The goal of the second phase of the project (and second cycle of the HCD process) was to create a low fidelity (Lo-Fi) prototype that would be used in a usability test with external test participants (see Figure 5.1).

The purpose of the Lo-Fi usability testing was to gather data and feedback in order to narrow the scope of the project and remove functionality deemed not necessary (or desired, by the user). The collected data and feedback would also be used to evaluate the proposed connection methods and the automatic free weight solution as a whole.

5.1 Meetings with stakeholders

As mentioned previously - it was at this point in the project, not fully understood how the CTS's underlying code/structure functioned. Therefore, a meeting with the stakeholders at Sony's R&D Center Lund Laboratory, together with Advagym stakeholders was set up. During the meeting, the user flow and connection methods from the concept ideation were proposed to the meeting's participants.

The proposed connection methods were deemed mostly feasible by R&D's stakeholders. The "Puck" and "Waypoint"-method required minimal additional implementation from their end - in regards to their CTS "backend" (underlying software and systems). The MQTT messages/broadcasts generated by the CTS broker would only need certain *filtration* - along with a fixed "waypoint" to be recognized by the system. The waypoint would be represented by fixed x, y and z coordinates in 3D-space.

When it came to the proposed "Gesture" connection method - that would require new detection and additional machine learning on their part. They therefore asked for evaluation of the proposed connection methods in the usability test - so as to determine if such an implementation was deemed necessary/desirable by the user.

In terms of the proposed user flow - the Advagym stakeholders approved of the idea of closely mimicking the current Advagym application. This would also make it easier for them to incorporate a free weight exercise tracking into their existing application. However, feedback was given when it came to the connection portion of the user flow.

Currently, Advagym users can just walk up to a gym machine and tap the puck situated there with their phone - which would then launch the corresponding exercise screen and let the user start working out. Therefore, each puck is closely "bound" or affiliated with a certain gym machine/exercise. How would the "free weight puck" work, in this regard?

At the end of the meeting, it was suggested that further meetings take place with engineers on the respective teams - when the Lo-Fi prototype usability tests had been completed - in order to aid in the following implementation of a Hi-Fi prototype.

5.2 Prototyping

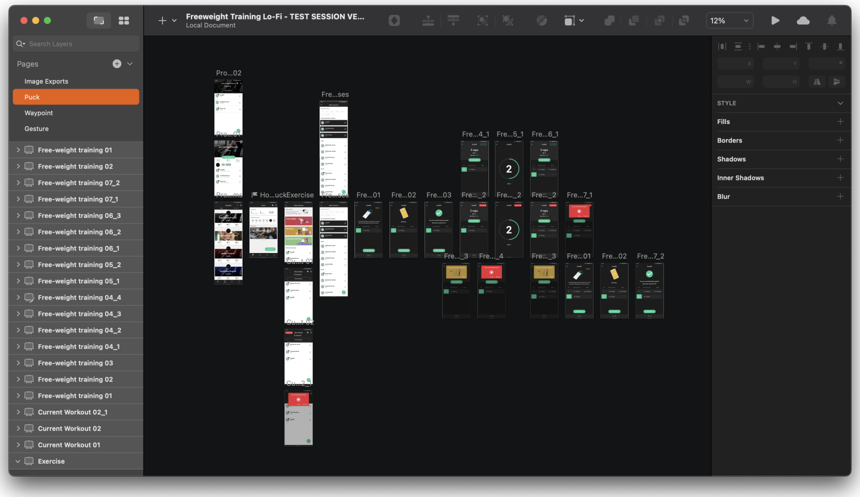


Figure 5.2: The Sketch vector graphics editor, with the Lo-Fi prototype project open.

Due to the COVID-19 pandemic and its subsequent restrictions and regulations in Sweden at the end of 2021 and early 2022, it was not possible to meet with potential usability test participants in-person. Therefore, the goal of the Lo-Fi prototyping was to create a digital smartphone prototype that test participants could use and interact with, over the internet. Since there were three different connection methods proposed - three distinct prototypes, each with different connection methods - would be presented to the test participants for evaluation (these are detailed below).

Sketch, a vector graphics editor for the macOS operating system (see Figure 5.2) has the functionality of letting users create interactable user interfaces/prototypes - which can also be shared over the internet with a simple Uniform Resource Locator-link (URL-link) [45].

With the help of several screenshots taken whilst using the current Advagym Android application (such as the ones seen in Figure 4.4) - work began to reconstruct that application as precisely as possible, within Sketch using its vector design tools. The different screens constructed were interactable in terms of being able to click/tap on certain user interface components, scrolling through list views and moving and navigating back-and-forth throughout the application. The screens and screen transitions were also animated so as to closely resemble the current Advagym application. The following screens were constructed (detailed below):

- Home screen
- Exercise Selection screen
- Free weight Selection screen
- Exercise screen

- Current Exercise screen
- Workout screen
- Program Selection screen
- Program screen

Home screen

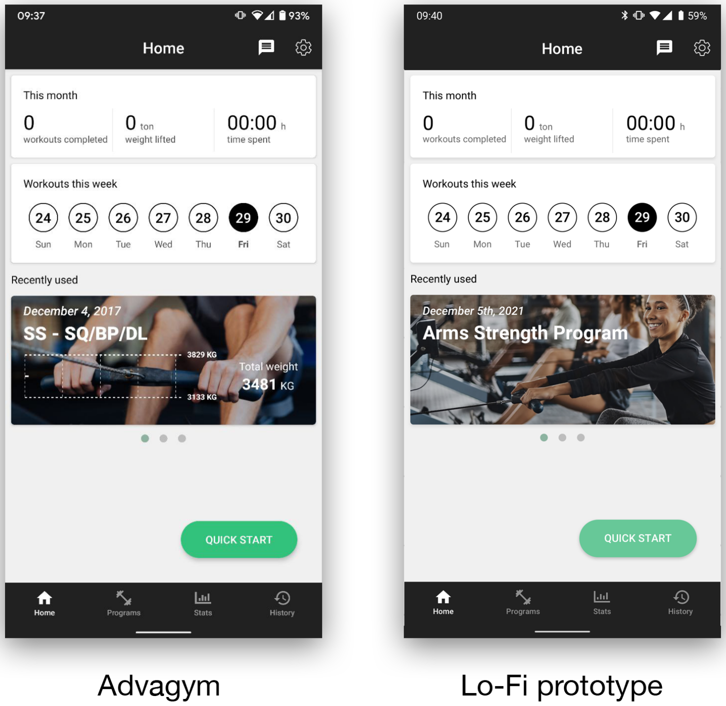
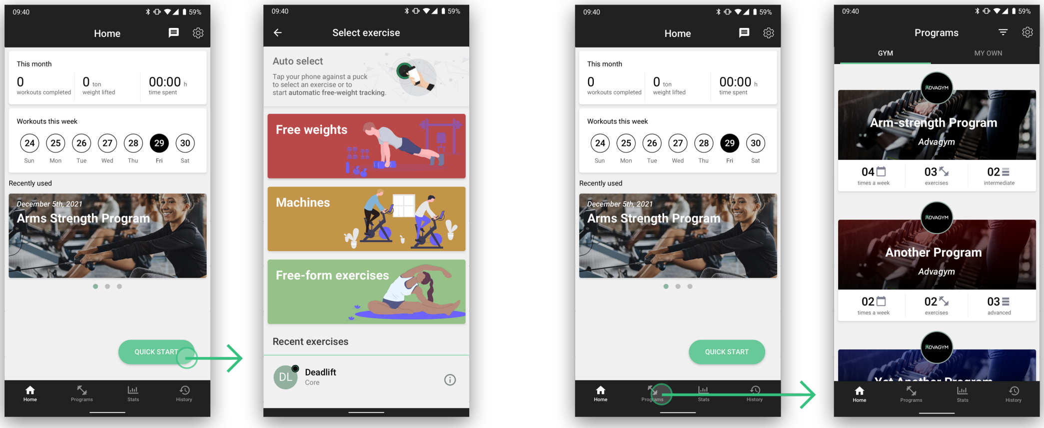


Figure 5.3: The "Home" screen of Advagym's Android application, followed by the Lo-Fi prototype equivalent.

This is the first screen that the user encounters when opening the application (see Figure 5.3). It serves as the top of the navigation hierarchy of the application, where the main navigable parts are shown on the dark *tab* at the bottom of screen - as well as at the very top-right through the use of icons (and labels) that signifies their action (for instance a cog-wheel for navigating to a Settings screen). Useful at-a-glance historical statistics are shown to the user in regards to their previous workouts - along with a direct link to a recently used workout program (i.e. a collection of exercises that a user performs during a single workout-session at the gym).



Home to Exercise Selection screen　　　　Home to Program Selection screen

Figure 5.4: Navigation result of the user either clicking/tapping on the "QUICK START" button or the "Programs" icon & label.

A button labeled "QUICK START" uses color- and size signifiers and affordances to inform the user that this is the main action available on the screen. Clicking/tapping that button takes the user to the "Exercise Selection" screen (see Figure 5.4). Another interactable part of the Home screen is the "Programs" icon (and label) on the bottom tab. If the user clicks/taps on that user interface (UI) component - the "Program Selection" screen is shown.

Exercise Selection & Free weight Selection screen

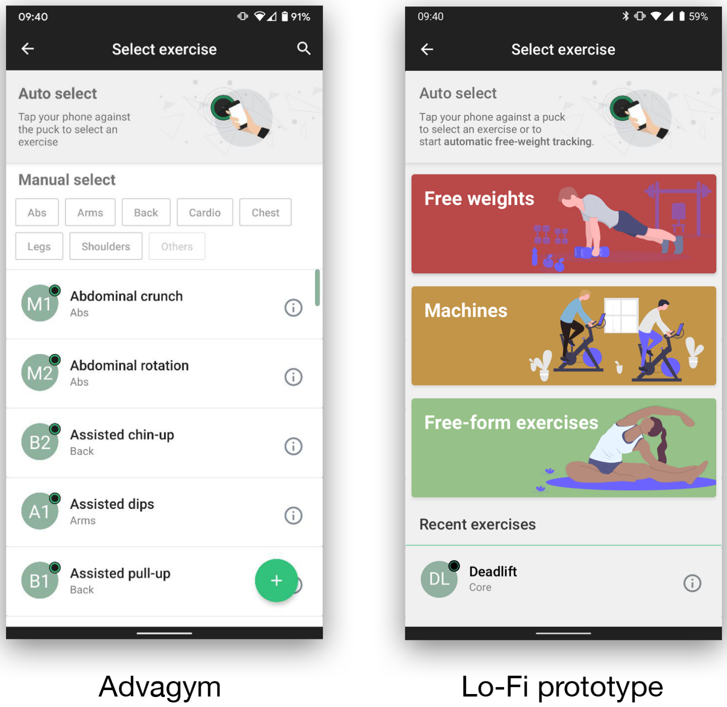


Figure 5.5: The "Exercise Selection" screen of Advagym's Android application, with the Lo-Fi prototype equivalent.

The Exercise Selection screen (see Figure 5.5) is shown to the user if they click/tap on the "QUICK START" button (or when clicking the Floating Action Button (FAB) in the Workout screen, detailed below). There is quite a large difference between the current Advagym application and what is proposed in the Lo-Fi prototype. The current Advagym application shows a long scrollable alphabetical list view of predominantly gym machine exercises that can be filtered depending on muscle group (search is also available through the magnifier button found in the top right of the screen). Since Advagym would be able to offer automatic free weight exercise tracking alongside their automatically logged gym machine exercises - along with perhaps future automatic free-form exercise tracking - the Lo-Fi prototype aims to change the user flow in regards to finding and choosing exercises.

The motivation behind the layout change was due to a previously-brainstormed *user story*: "When a user enters the gym and wants to start an exercise of some kind - what overarching types of exercises/activities are there? How can we guide or help the user?". Instead of having one long scrollable list (that could be quite daunting for an inexperienced gym goer) - the entire gym experience is now split into three main "categories" in regards to exercise types available in a gym: *free weights*, (gym) *machines* and *free-form exercises*. This change would hopefully make it easier for the gym-goer to find and discover exercises pertaining to the type of equipment (or lack thereof) they want to use during their workout.

At the top of the screen, under the status bar (with the screen name and dark background) - is a small panel with an illustration of a phone tapping a puck along with some text (instructions). In the current Advagym application - the text tells the user that they can just tap a puck with their phone in order to automatically navigate the user to that gym machine's exercise screen. In the "Lo-Fi prototype 3" that uses the puck as a connection method, this pane is also shown with slightly altered text and functionality (detailed below in Section 5.2.3).

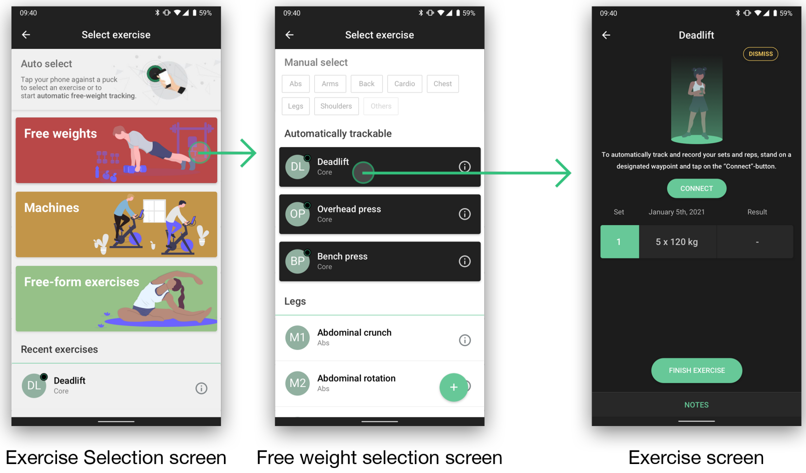


Figure 5.6: Navigation result of the user either clicking/tapping on the "Free weights" category, and then clicking/tapping the "Deadlift" exercise.

Clicking/tapping on the free weights category will take the user to the Free weight Selection screen that closely resembles the current Advagym application's Exercise Selection screen (see Figure 5.6). When the user has found an exercise that they want to perform, a simple click/tap on that exercise will bring them to the corresponding Exercise screen. Automatically trackable exercises are afforded a darkened background (raised, with a drop shadow) so as to signify their difference from other non-automatically-trackable exercises. A FAB with a plus-sign in the middle of it is also available in the bottom-right of the screen. Clicking that button will let the user create their own custom exercise.

Exercise screen

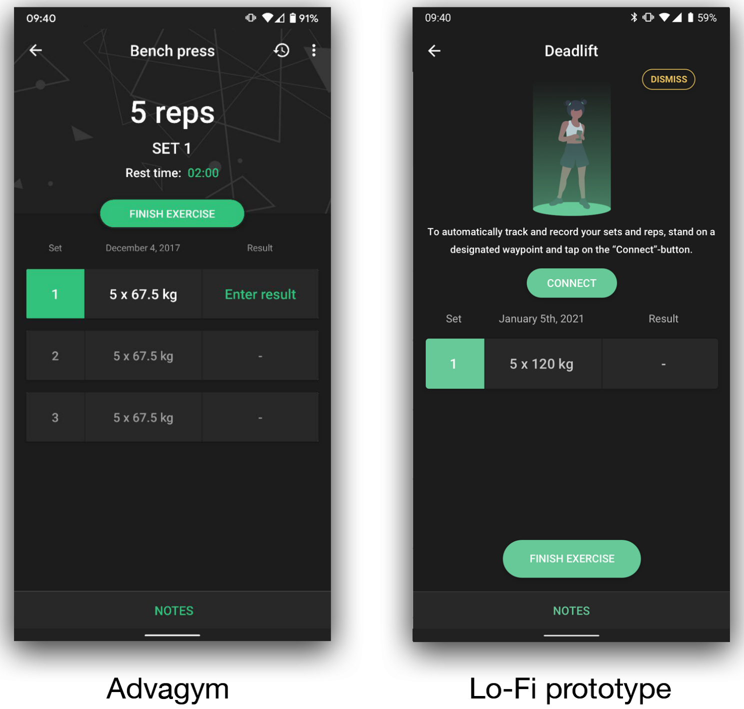


Figure 5.7: The "Exercise" screen of Advagym's Android application, with the Lo-Fi prototype equivalent.

This is the screen (see Figure 5.7) that is shown when the user has clicked on a specific exercise in the Exercise Selection screen (or the Workout screen, detailed below). This screen functions as the user's overview of the current exercise with its list of sets, repetitions and weights. This screen was chosen as the location for the CTS connection *onboarding* (a user flow that gives the user a guided introduction) and the main point of CTS system error feedback.

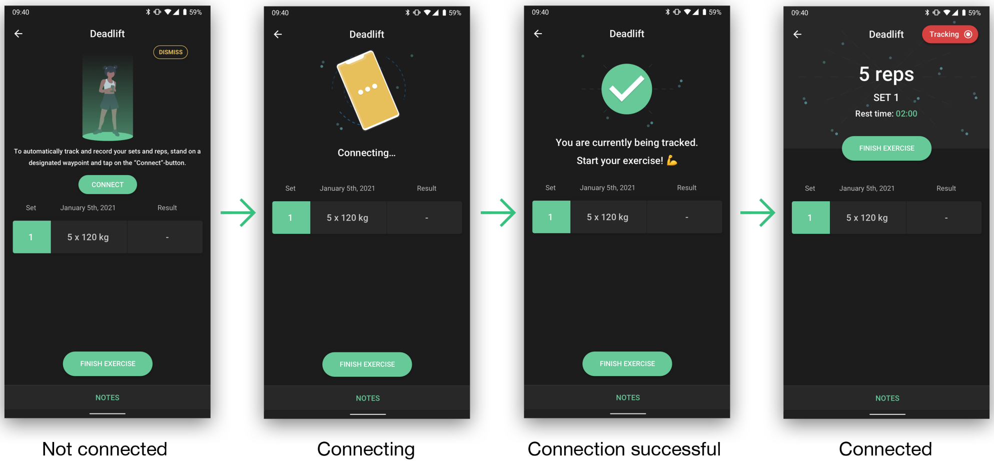


Figure 5.8: The four connection states shown in the Exercise screen.

Depending on the connection method (i.e. Prototype 1, 2 or 3) - the area with the illustrations at the top of the screen will change its appearance. The illustrations and accompanying text (and/or button) will help inform the user of how to connect to the CTS. In order to provide the user with feedback of how the CTS connection attempt is progressing, the screen will also change its appearance (specifically the top area with its illustrations) - depending on the following four connection states (see Figure 5.8 where the Waypoint connection method is shown):

- Not connected
- Connecting
- Connection successful
- Connected

If the connection is unsuccessful, the connection state will return to the "Not connected"-state.

As seen in the last screen, in Figure 5.8 - there is a red elongated button in the top-right corner of the screen. This is known as the "Connection status" button. Several iterations later - the button had been afforded a "Tracking"-label, recording icon and a red color in order to clearly indicate to the user that something important is happening. If the user is connected to the CTS - the red connection status button is shown.

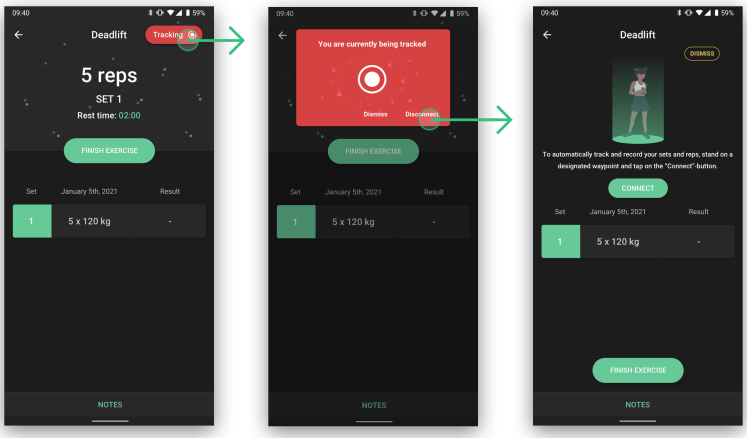


Figure 5.9: Result of the user clicking/tapping the connection status button and choosing the "Disconnect" option in the pop-up window.

If the user wants to disconnect from the automatic tracking - clicking/tapping on the connection status button will display a pop-up window that explains the current connection status to the user - along with options to either dismiss the pop-up window, or disconnect from the automatic tracking (see Figure 5.9). Clicking the disconnect button returns the screen to the "Not connected" state layout and CTS connection onboarding illustrations.

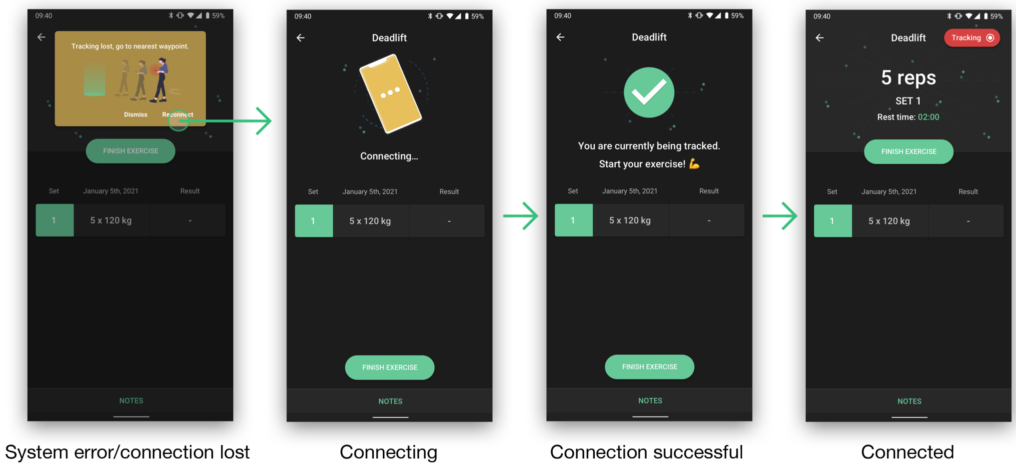


Figure 5.10: Result of the user clicking/tapping the "Reconnect" option in the pop-up window.

If a system error occurs whilst the user is connected; such as the system losing track of the connected user - a pop-up window will show up with instructions on how to reconnect. If the user chooses to reconnect, the screen changes its appearance to the different connection states (see Figure 5.10 where the Waypoint connection method is shown).

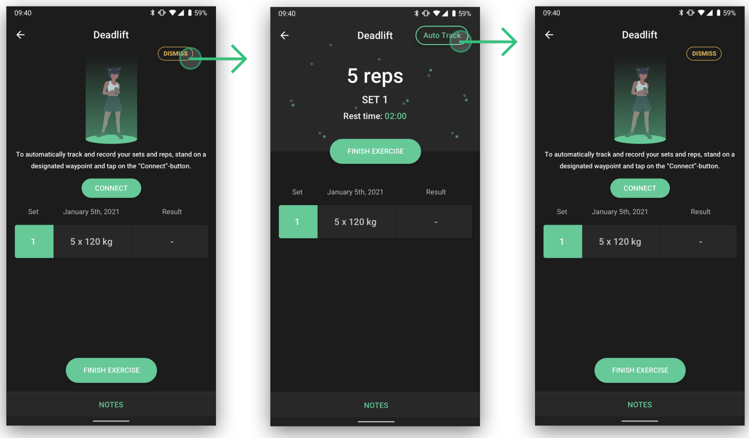


Figure 5.11: Result of the user pressing the "Dismiss" button, followed by the user clicking on the "Auto Track"-labeled button.

If the user does not want to use automatic free weight exercise tracking - clicking the yellow "Dismiss" button, will result in the Exercise screen changing its appearance - and will now let the user manually log sets, reps and weight (see Figure 5.10). This screen appearance closely resembles the "connected" state - but does not have a red connection status button, indicating an ongoing connection/the user currently being tracked. If the user wishes to use the automatic free weights tracking feature - pressing the green "Auto Track"-labeled button returns the screen to its original layout and CTS connection onboarding illustrations and instructions.

Current Exercise screen

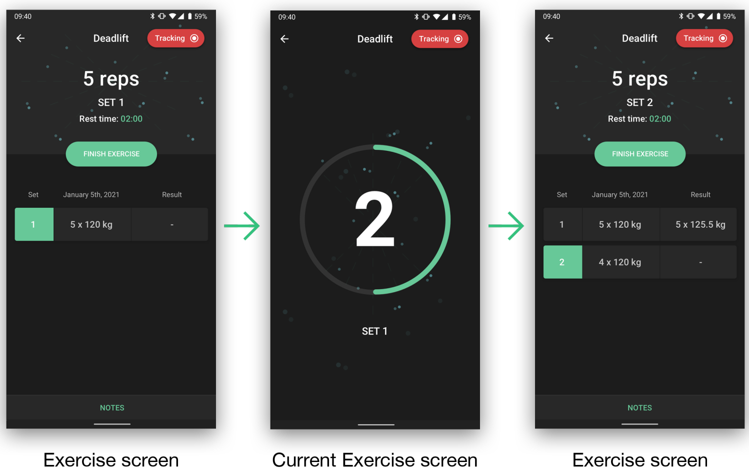


Figure 5.12: The automatic transitions of the Exercise and Current Exercise screens.

When a user starts performing the exercise corresponding to the exercise chosen (the name of the chosen exercise is found at the top of the Exercise screen) - the Current Exercise screen is automatically transitioned to (see Figure 5.12). With every repetition that the user performs, a large counter in the middle of the screen is incremented. When the user is done with their set, the Current Exercise screen automatically transitions back to the Exercise screen. The Exercise screen's list of sets is now updated to reflect the performed set and its number of reps (shown in the last screen in Figure 5.12).

Workout screen

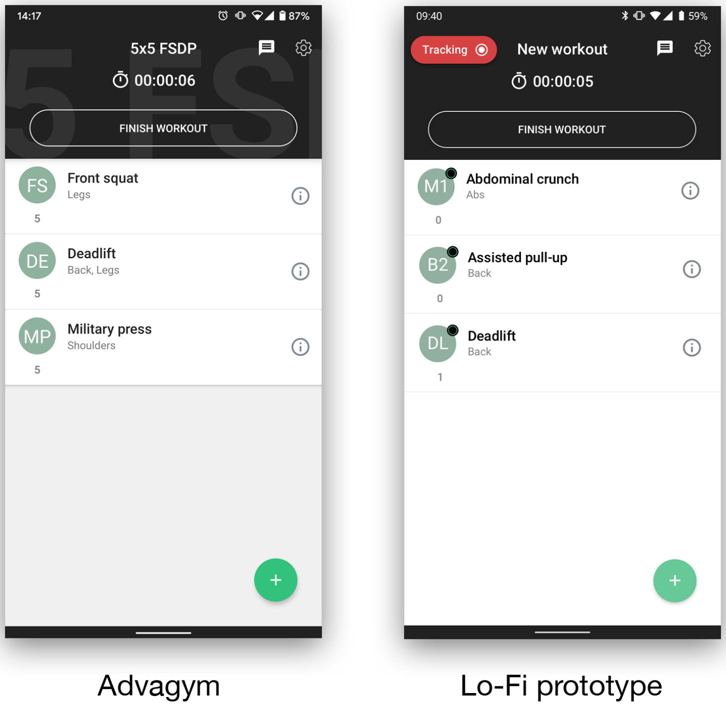


Figure 5.13: The "Workout" screen of Advagym's Android application, with the Lo-Fi prototype equivalent.

All of the exercises that a user of Advagym's current application records and logs during their workout (i.e. the entire gym session) - is found in a central "Workout" screen (see Figure 5.13). The Lo-Fi prototype is almost identical in this regard - with one important distinction: it features the same connection status button found in the Exercise screen. This is shown when the user is still tracked and connected to the CTS. The same exact functionality also applies here (i.e. the user can manually disconnect from the tracking if they so choose).

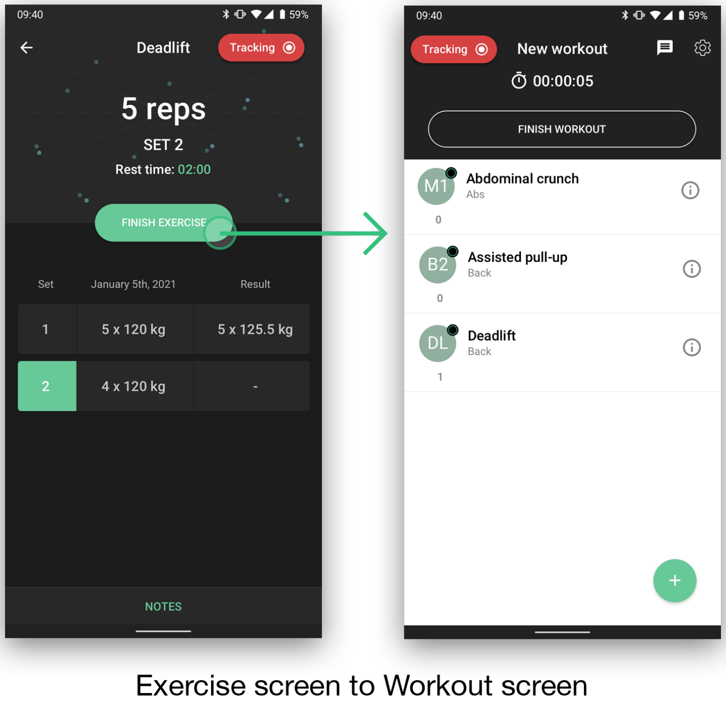


Figure 5.14: Result of the user clicking/tapping the "FINISH EXERCISE" button in the Exercise screen.

To get to the Workout screen - the user clicks/taps the "FINISH EXERCISE" button, or the back button located in the top left corner of the Exercise screen (see Figure 5.14). However, they first need to have logged at least one set of any exercise - otherwise the user is returned to the Exercise Selection screen (since the user most likely chose the wrong exercise and/or hasn't done any prior exercises and wants to find the correct one).

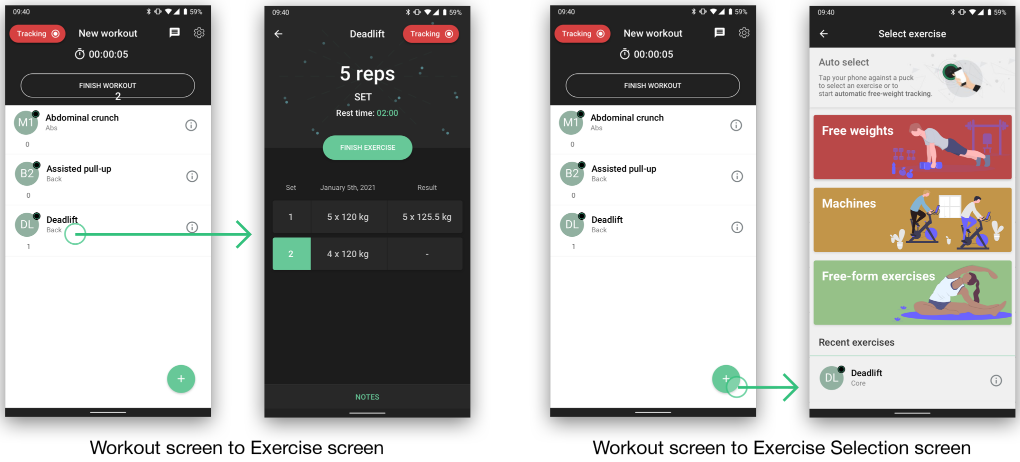
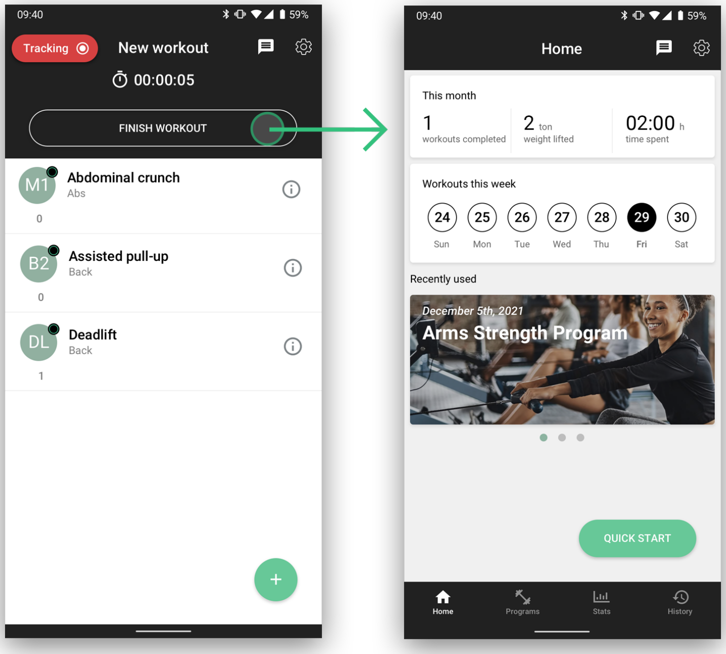


Figure 5.15: Result of the user clicking/tapping an exercise in the Workout screen - followed by the result of the user clicking/tapping the FAB in the Workout screen.

If the user wants to get back to the Exercise screen of a certain exercise found in the workout - they can simply click/tap that exercise in the exercises list (see first pair of screens in Figure 5.15). If the user would like to find a *new* exercise - they can click/tap on FAB in the lower-right corner of the screen. This navigates the user to the Exercise Selection screen (see second pair of screens in Figure 5.15).



Workout screen to Home screen

Figure 5.16: Result of the user clicking/tapping the "FINISH WORKOUT" button.

When the user is done with all of their exercises and is ready to leave the gym/does not wish to continue to use the Advagym application - they simply click/tap on the "FINISH WORKOUT" button in order to return to the Home screen (see Figure 5.16). In Advagym's current application - this also implies that the user registers and saves their completed workout to their account - which can then be referenced in terms of historical statistics and trends.

Program Selection & Program screen

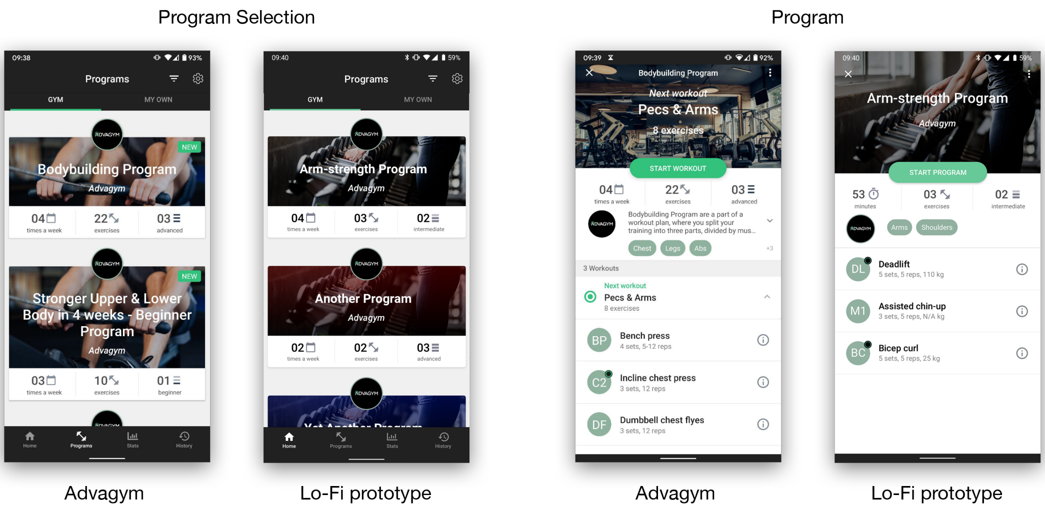


Figure 5.17: The "Program Selection" followed by the "Program" screen of Advagym's Android application, with the Lo-Fi prototype equivalents.

The last screens and features of the Lo-Fi application mimics that of Advagym's "Programs" features (see Figure 5.17). A program is a predefined list of exercises with a specific amount and sets and reps within each exercise. This is aimed at guiding the user through a workout session at the gym. Users can also create custom programs in order to easily plan their visits to the gym ahead of time. Clicking/tapping on a program in the Program Selection screen navigates the user to the Program screen of that particular program.

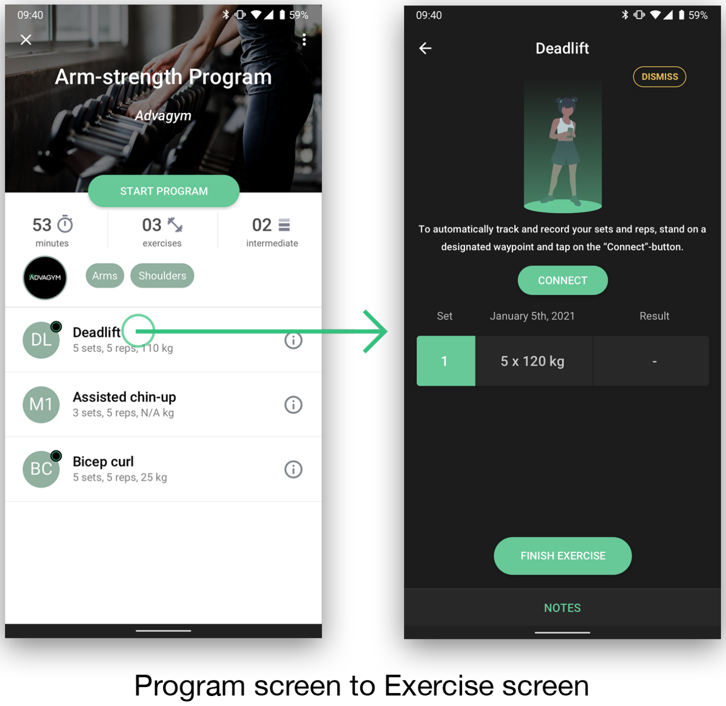


Figure 5.18: Navigation result of the user either clicking/tapping on the "Deadlift" exercise.

When the user wants to start the program, they can either click the "START PROGRAM" button, or simply click on the exercise that they would like to start with (see Figure 5.18). After that, the user flow is identical to the above-described user flow of using the "QUICK START" button on the Home screen and going through the exercise selection. One difference is that when the user clicks the "FINISH EXERCISE" button or the back button in the Exercise screen - they are taken to the Workout screen that already has an exercise list populated with the exercises found in the chosen program's exercise list.

5.2.1 Lo-Fi prototype 1: Gesture

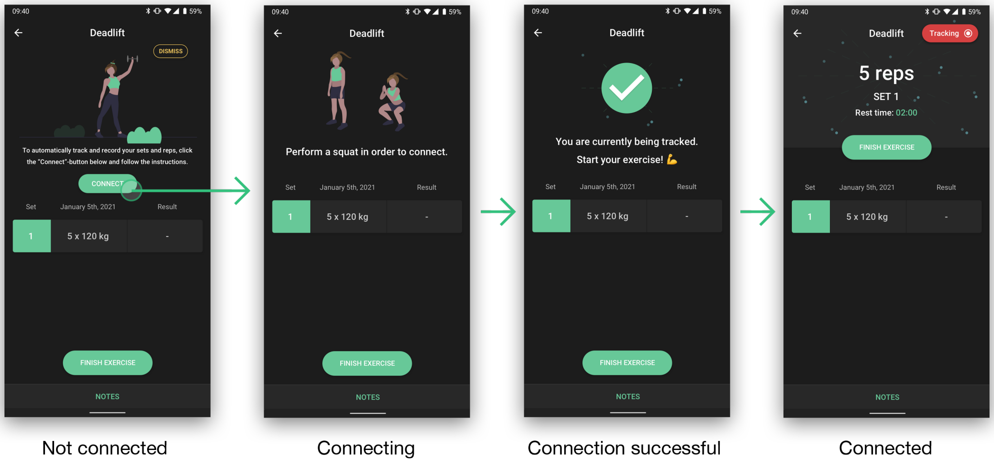


Figure 5.19: Connection states when connecting to the CTS using a gesture.

The first of the three Lo-Fi prototypes was specifically created to test and evaluate a "Gesture" connection method. As previously detailed - the method implies that the user performs a specific gesture, which the CTS is able to detect - and can therefore link the tracked skeleton with the user performing the gesture. The gesture itself could be anything, such as: the user squatting down, or having the user raise their arms above their head.

The connection states and user flow is shown in Figure 5.19. When the user presses the "CONNECT" button in the Exercise screen, they are prompted to perform a certain gesture. In order for the system not to get false-positive user-skeleton linking, the gesture that the user is prompted to perform will be randomized. When the CTS has successfully detected a skeleton performing the prompted gesture - the user-skeleton link will be established and the user will be connected. In that case, the user is shown a connection successful illustration which is then followed by the connected state of the Exercise screen - displaying the red connection status button.

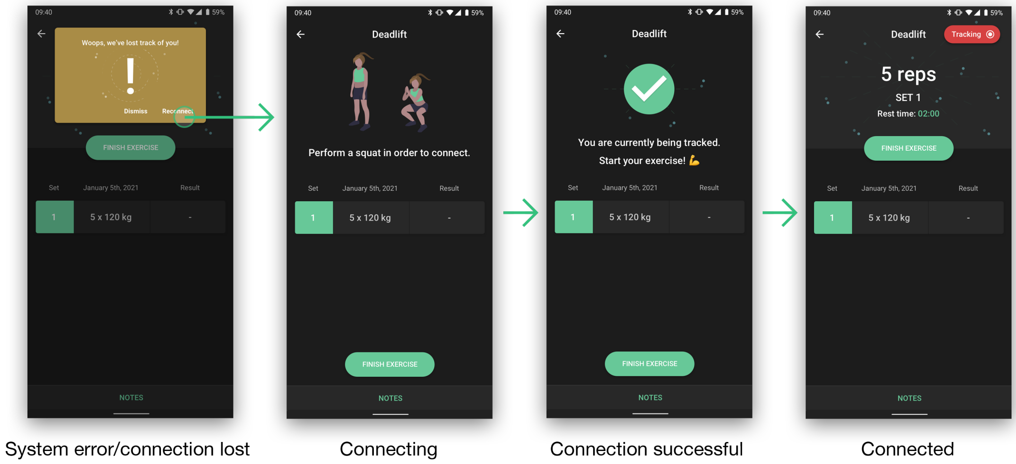


Figure 5.20: Result of the user clicking/tapping the "Reconnect" option in the pop-up window using the gesture connection method.

If a system error occurs; such as the system losing track of the connected user - a pop-up window will show up with instructions on how to reconnect (see Figure 5.20). In the case of this prototype - upon clicking the "Reconnect" option in the pop-up window - the user is once again prompted to perform a specific gesture. If the connection is successful, the "connected" state of the Exercise screen is shown to the user.

5.2.2 Lo-Fi prototype 2: Waypoint

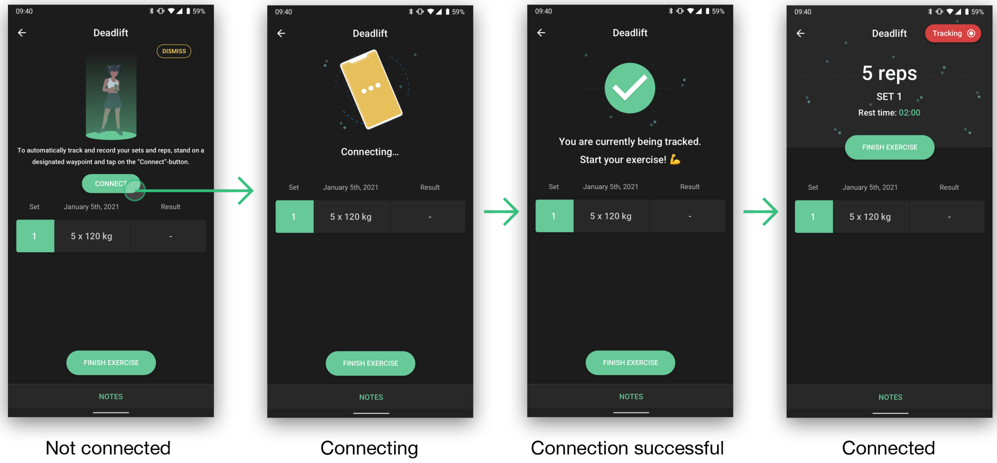


Figure 5.21: Connection states when connecting to the CTS using a waypoint.

The second of the three Lo-Fi prototypes was specifically created to test and evaluate the "Waypoint" connection method. As previously detailed - the method implies that the user walks over to a specific location in the gym - that is clearly marked with the help of signage, paint or some other material/objects. Since the CTS has the exact X, Y and Z coordinates of the waypoint stored in its system - the CTS is able to determine what skeleton is "inside" of the specified waypoint boundary. The tracked skeleton can therefore be linked to the user.

The connection states and user flow is shown in Figure 5.19. The illustration and instructions presented in the "not connected" state of the Exercise screen instructs the user to find and stand inside the designated waypoint. At the precise moment of the user pressing the "CONNECT" button, the system tries to find a skeleton that is standing inside the waypoint boundary. If the CTS is successful in doing so - the user-skeleton link is established and the user is connected. The Exercise screen will then move to the connected state, and will display the red connection status button.

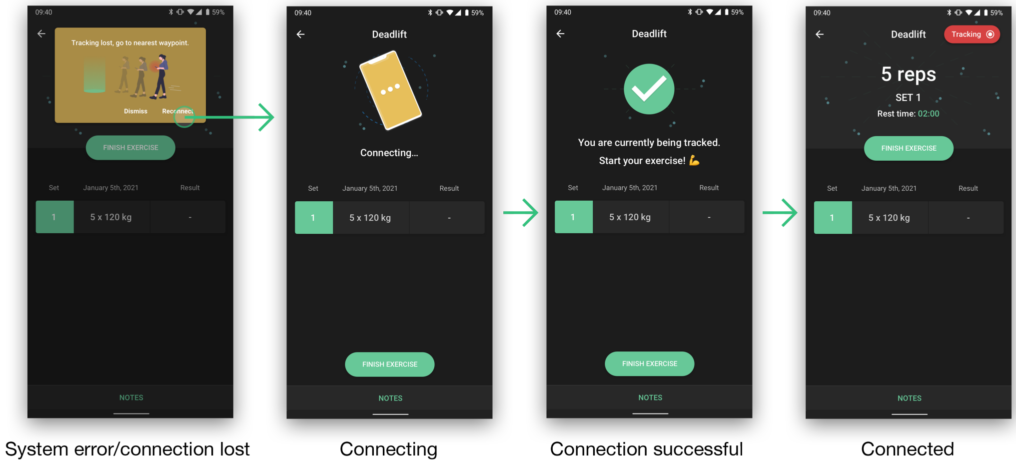


Figure 5.22: Result of the user clicking/tapping the "Reconnect" option in the pop-up window using the gesture connection method.

If a system error occurs; such as the system losing track of the connected user - a pop-up window will show up with instructions on how to reconnect (see Figure 5.22). In the case of this prototype - the user is instructed to once again walk over to the waypoint. Once there, clicking the "Reconnect" option in the pop-up window will trigger a connection attempt. If the connection is successful, the previous connected state of the Exercise screen is shown to the user.

5.2.3 Lo-Fi prototype 3: Puck

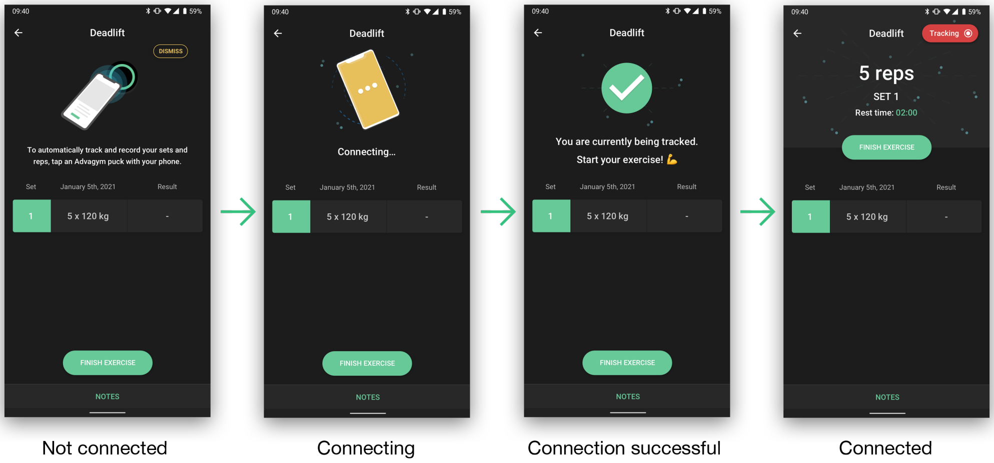


Figure 5.23: Connection states when connecting to the CTS using a puck.

The last of the three Lo-Fi prototypes was specifically created to test and evaluate the "Puck" connection method. As previously detailed - the method implies that the user taps a specific puck in the gym with their phone. In order to differentiate the puck from the other pucks placed on gym machines - there might be additional signage accompanying the "free weight connection puck" (or affording the puck a different color signifying the different use case).

The only difference between the "Waypoint" prototype and the "Puck" prototype is what initiates the connection attempt. In the case of the Waypoint, the user has to click a "CONNECT" button. Conversely, the "Puck" prototype uses the tapping of the user's phone on the puck as the initiation. Both prototypes utilize the waypoint X, Y and Z coordinates stored within the CTS and share the same connection method.

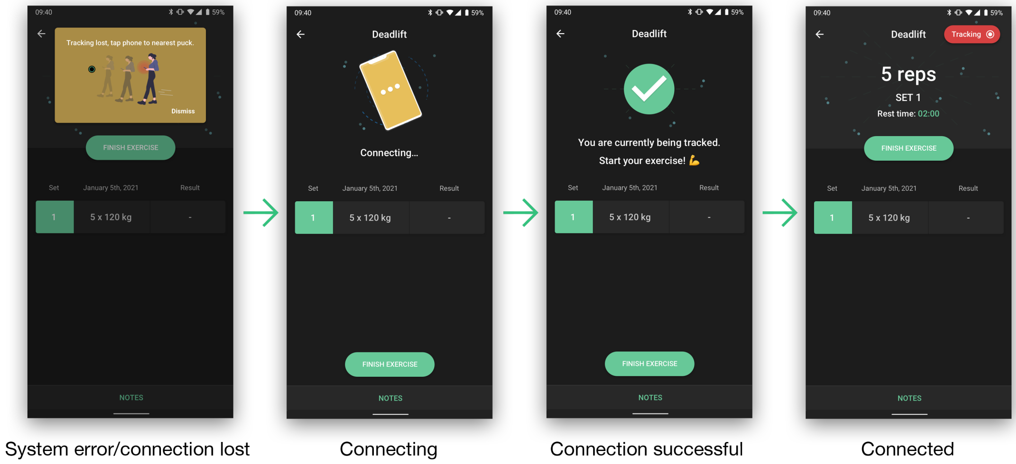


Figure 5.24: Result of the user tapping the free weight connection puck with the "tracking lost" pop-up shown.

If a system error occurs; such as the system losing track of the connected user - a pop-up window will show up with instructions on how to reconnect (see Figure 5.24). In the case of this prototype - the user is instructed to once again tap a puck. Tapping a puck with their phone will trigger a connection attempt. If the connection is successful, the previous connected state of the Exercise screen is shown to the user.

5.3 Lo-Fi usability test

When the interactions and functionality of all of the three Lo-Fi prototypes had been implemented with the help of Sketch's prototyping tools - it was time for the project's first usability test. The goal of the usability test was to evaluate which of the three proposed connection methods showed the most promise in regards to quantitative and subjective metrics.

This would also allow for a narrowing of the project's overall scope, in order for the real-world implementation of the Hi-Fi prototype to be feasible in regards to time and resource constraints mentioned previously (in Section 5.1). In addition - the results of the test will be used to answer the research questions of the thesis, help with sub-goals of the project and also infer what iterations and testing methodology is needed for the next phase of the thesis work.

In order to carry out the usability test, a test plan was first created which detailed the methodology of the test, the data being collected, what resources were required and how the test participants would be chosen/briefed. This test plan was reviewed and approved by both project supervisors.

5.3.1 Test design

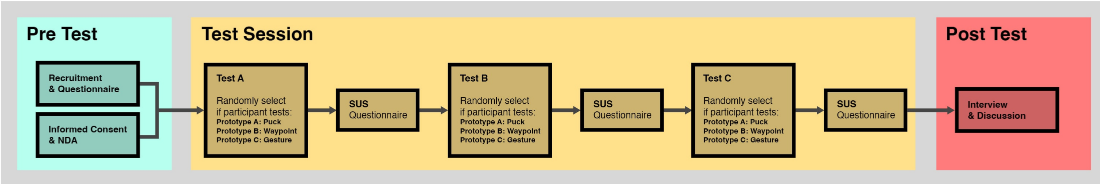


Figure 5.25: The different parts and order of tasks, of the Lo-Fi usability test.

The test was carried out in three parts: "Pre Test", "Test Session" and "Post Test". The different parts were carried out in a specific order with the Pre Test being the first - followed by a Test Session and lastly a Post Test (see Figure 5.25).

Pre Test

This part/phase of the test consisted of the recruitment, and initial contact with the test participants. The participants were able to sign up for the test with the help of an online questionnaire that also served as an opportunity to gather some background information regarding the participant.

An online questionnaire was chosen as the facilitator of the first phase of the test due to the flexibility it provided along with how the response data was recorded. Participants could easily be recruited through the use of a single static URL-link - and they could complete the questionnaire when they saw fit. The data and background information supplied by the participants were easily quantifiable due to the tools available within Google Forms (the platform of choice for the questionnaire).

The questions posed in the questionnaire were used in ideation and iterative processes, along with establishing the training habits/patterns and exercise/gym knowledge of the participants. Such questions include: "How many times per week do you exercise?" and "How long have you kept up with your training frequency?". The questionnaire is found in Appendix A.

The Pre Test allowed the test moderator to brief the participant of the test and its background. The participants also needed to sign a Non-disclosure Agreement (NDA) and informed consent, to be able to continue with the test - since the systems, solutions and technologies used in the thesis work were not all public.

Test Session

Due to the previously mentioned nature of the COVID-19 pandemic and its subsequent restrictions and regulations in Sweden, all test session parts of the test was conducted through the use of video conference and screen-sharing software over the internet (in this case, Google Meet [52]).

At the start of the test session - the participants were asked to imagine being in a gym environment that had an area for filling up water, gym machines and a free weight training area. Participants were also asked to place a smartphone in front of them. Depending on what prototype was used, a further description of the gym area and what objects/things were available, was presented. When the participant was ready, they were asked to perform specific tasks when using the three proposed Lo-Fi prototypes:

- Lo-Fi prototype 1: Gesture
- Lo-Fi prototype 2: Waypoint
- Lo-Fi prototype 3: Puck

The order in which the prototypes were presented to the test participant was randomized. This was done in order to not introduce unwanted bias in the test and to also evaluate if there is any correlation between the order the prototypes are shown in, with regards to results. That meant that the participants of the tests carried out one of the following **test scenarios** (combinations of the three prototype tests):

- Test A: Gesture → Test C: Puck → Test B: Waypoint
- Test A: Gesture → Test B: Waypoint → Test C: Puck
- Test C: Puck → Test B: Waypoint → Test A: Gesture
- Test C: Puck → Test A: Gesture → Test B: Waypoint
- Test B: Waypoint → Test C: Puck → Test A: Gesture
- Test B: Waypoint → Test A: Gesture → Test C: Puck

For each test scenario (and its accompanying prototype), the order of the tasks that the participant needed to complete were as follows:

1. Ask the participant to begin the process of doing a deadlift exercise through the application.
2. Ask the participant to connect to the tracking system.
3. Ask the participant to begin an exercise.
4. Ask the participant to "walk away to get some water" outside of the imagined free weights area.
5. *The system has now lost contact with the participant.*
6. Ask the participant to reconnect to the tracking system.
7. Ask the participant to disconnect from the tracking system, manually.
8. Ask the participant to reconnect to the tracking system.
9. Ask the participant whether they are connected to the system or not.

After each and every individual prototype tested (Prototype 1, 2 or 3) in the chosen test scenario, the test participant was asked to fill out a SUS questionnaire regarding that specific prototype. This meant that each test participant filled out three separate questionnaires (the questionnaires have identical questions which are found in Appendix B).

Post Test

Lastly, a semi-structured interview was conducted, where the following questions were asked in order to collect some quantitative and qualitative data:

1. Have you used any gym training application before? If so: which one(s)?
2. Using the application and the tracking system - which of the 3 prototypes (puck, waypoint, gesture) and their connection methods, would you prefer?
3. Having tested the application - would you like to use a system that tracks your exercises, like the one proposed? Why, why not?
4. Any other comments?

If there were any other questions, thoughts or ideas that came to mind whilst observing the participant completing tasks during the test session - they were also asked in this portion of the test.

5.3.2 Data collection

Quantitative and qualitative (subjective) metrics and data were collected throughout the duration of the test. Additional data about the participants was collected from the online questionnaire that was sent out in advance of the test session. The preceding questionnaire was specifically aimed to analyze the participants' existing training habits; along with its frequency, intensity and what type of exercises the participants perform in their daily life. Additional data concerning impairments of different kinds (i.e. vision, mechanical) was also collected.

During the test session, the participants' interactions with the prototypes were video recorded along with the audio of the voice call. The resulting metrics with their associated fail/pass criteria was recorded and included the following:

- **Success**: task was completed with no assistance required within a reasonable amount of time depending on the task.
- **Non-critical error**: task was completed with minimal assistance, such as a minor hint and/or restating the task objective.
- **Critical error**: task was not completed or was completed with major assistance.
- **Time on task**: the amount of time each individual task took for the test participant to complete.

Additionally, the results of the SUS questionnaires filled out by the test participants were used to create SUS scores and the mean SUS score for each prototype and its connection method (using Equation 3.1). In order to evaluate the usability the CGS was used to provide letter grades for the different prototypes (see Table 3.1).

Lastly, subjective data from the participants' answers to the Post Test interviews was collected, to determine if the participants understood the system/prototypes - or not.

5.3.3 Resources

As noted above - due to the COVID-19 pandemic - the tests were not conducted in-person in Lund at the Sony offices. The tests were performed via online methods. Therefore, the only tools and resources needed were:

- The three Lo-Fi prototypes, shared to the test participants via Sketch's prototyping tools.
- A computer equipped with a webcam and a microphone.
- Recording software that could record the screen, webcam and audio of the participants during the test session.

5.3.4 Test participants

The first test participant acted as a pilot test for the proposed usability test. The pilot test led to parts of the test being fixed/iterated upon - that did not work correctly (such as two bugs found in the Sketch's prototype testing tools). After that, 12 test participants were chosen to complement the number of prototypes being tested: since there were 6 test scenarios, each scenario could then be conducted twice.

User group

Since Advagym's user group is broad and diverse there were no predefined user groups targeted. The sources used for the recruitment of test participants was therefore diverse and varied.

The only restrictions imposed in the recruitment process was that the test participants were able to physically complete simple tasks involving a smartphone, computer and an object of their choice. The test participants also needed to be able to see/interpret information and the user interfaces displayed on their computer screen.

Recruitment

Recruitment of the test participants was conducted through a student Facebook page and friends and families of Advagym's employees and the project supervisors.

Attended participants

As previously mentioned - 12 test participants were recruited and carried out the usability test.

The data and insights gained from the recruitment questionnaire (found in Appendix A), was used for future Hi-Fi ideation and iteration. It was however *not* used to evaluate the Lo-Fi prototypes. This, due to the fact that *no* correlation was found between the test participants gender, age, exercise activity, gym experience, etc. - and the usability of the prototypes.

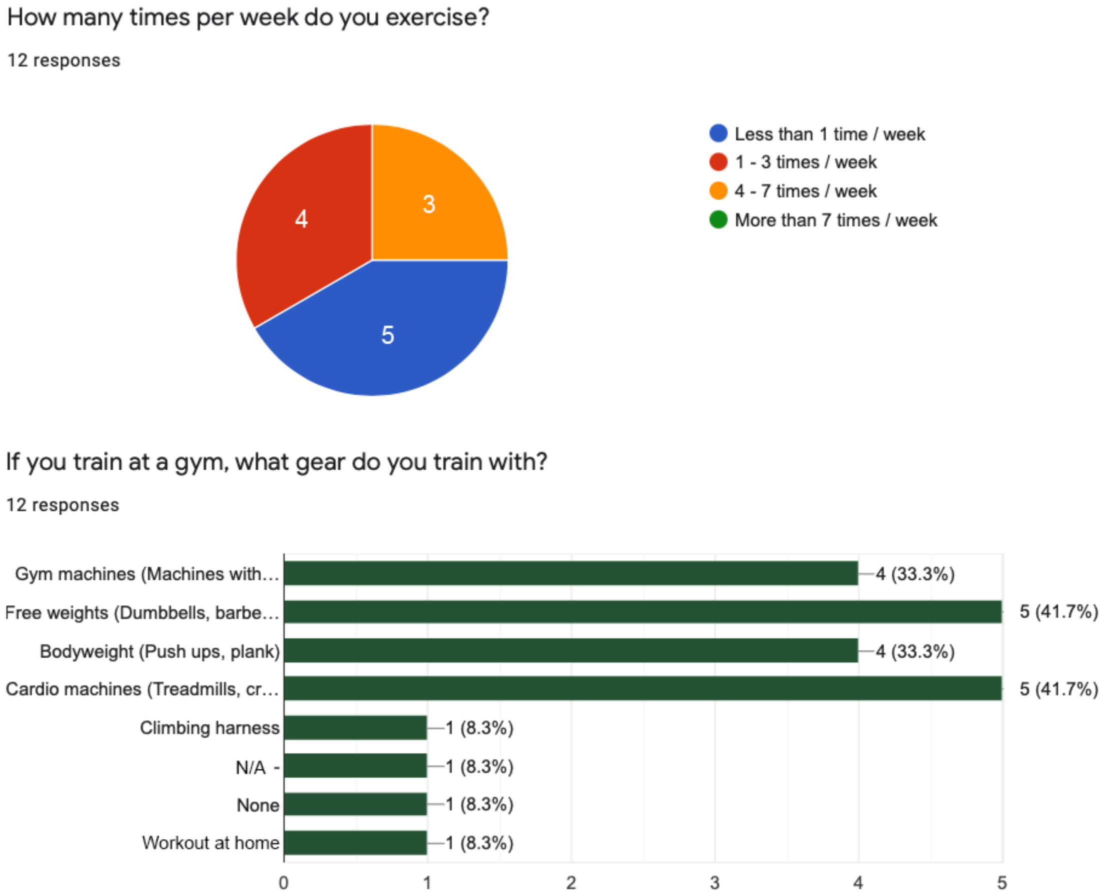


Figure 5.26: Distribution of exercise activity during a week, along with the distribution of equipment used/exercises performed when attending a gym.

The gender distribution of the participants was 9 males and 3 females. The average age of the participants was 27, within the span of 25 to 30 years old. The exercise activity of the test participants, along with what equipment/exercises they used when going to gym are found in Figure 5.26.

5.3.5 Procedure

Table 5.1: The required material, time estimates and descriptions for each part and step of the Lo-Fi usability test.

Part	Step	Description	Material	Time
Pre Test	Sign Up / Online Questionnaire	To register for the test, participants will fill out an online questionnaire.	Online Questionnaire	N/A
Pre Test	Briefing	Receive participant. Participant signs NDA. Brief participant of the structure and background of the test	- Testing script - Non Disclosure Agreement	5-10 min
Test Session	Test A	Ask the participant to complete the tasks associated with the tested prototype (Prototype A, B or C).	- Testing script - Recording software	4-7 min
Test Session	Test B	Ask the participant to complete the tasks associated with the tested prototype (Prototype A, B or C).	- Testing script - Recording software	4-7 min
Test Session	Test C	Ask the participant to complete the tasks associated with the tested prototype (Prototype A, B or C).	- Testing script - Recording software	4-7 min
Post Test	SUS questionnaire	The questionnaire will let the participant score the usability of the user interface/system.	SUS questionnaire	2-5 min
Post Test	Short interview	A couple of questions will be asked where the participant is able to summarize their experience and provide some additional quantitative/qualitative data.	- Testing script - Recording software	2-5 min
			Total time:	~40 min

The exact procedure, materials required - along with the time estimates for the different parts and the total estimated duration of the test are detailed in Table 5.1. A testing script was also used in order for the test moderator to accurately reproduce the test for each participant. The Lo-Fi prototype usability test script can be found under Appendix C.

5.3.6 Results

Test metrics

Table 5.2: The results and metrics for each part/task concerning the Lo-Fi prototype test scenarios.

Test A: Gesture				
Step	**Success**	**Critical Errors**	**Non-Critical Errors**	**Time on Task (mean, seconds)**
Start doing a deadlift exercise	11	0	1	15s
Connect to auto-track system	12	0	0	21s
Begin and complete exercise	12	0	0	11s
Reconnect to auto-track system	12	0	0	13s
Disconnect from auto-track system	12	0	0	14s
Reconnect to auto-track system again	12	0	0	9s
Test B: Waypoint				
Step	**Success**	**Critical Errors**	**Non-Critical Errors**	**Time on Task (mean, seconds)**
Start doing a deadlift exercise	12	0	0	17s
Connect to auto-track system	9	0	3	32s
Begin and complete exercise	12	0	0	12s
Reconnect to auto-track system	11	0	1	16s
Disconnect from auto-track system	12	0	0	13s
Reconnect to auto-track system again	12	0	0	12s
Test C: Puck				
Step	**Success**	**Critical Errors**	**Non-Critical Errors**	**Time on Task (mean, seconds)**
Start doing a deadlift exercise	12	0	0	13s
Connect to auto-track system	12	0	0	22s
Begin and complete exercise	12	0	0	12s
Reconnect to auto-track system	11	0	1	15s
Disconnect from auto-track system	11	0	1	16s
Reconnect to auto-track system again	12	0	0	8s

Every single test participant was able to complete all tasks given during the usability tests (i.e. there were no "critical errors", see Table 5.2). There were however cases where the test participant received minor assistance in the form of a hint or simply restating the task given; that were regarded as a "non-critical error". The "Waypoint" prototype had the most non-critical errors. Both the "connect to auto-track system"- and the "reconnect to auto-track system"-tasks related to that prototype had a shared error source in all test participants: the test participant was not able to recall the background information previously given, about there being a marked spot on the floor in the gym's free weight area.

The "Time on task"-metric did not offer any clear insight into which of the prototypes was better than the other in terms of completion time. The only deviation was that of the Waypoint prototype's "connect to auto-track system"-task where 3 test participants' longer completion time drove up the mean time on task in comparison to the other prototypes.

SUS scores

Table 5.3: The resulting mean SUS score (out of 100) from the SUS questionnaires for each prototype.

Prototype	SUS Score (mean):
Puck	**83.5**
Waypoint	**84.1**
Gesture	**84.7**

36 different SUS-scores were compiled from the 3 prototypes tested and the 12 test participants. By comparing the mean SUS-scores to the corresponding range found in the CGS table (see Table 3.1), a grade to describe the usability of the prototype was arrived at [50]. The mean SUS-scores were very similar (see Table 5.3), with the "Waypoint" prototype scoring a **84.1** whilst the "Gesture" prototype scoring **84.7** - earning both an ***A+*** usability grade. The "Puck" prototype however, placed last with a **83.5** score, thereby earning it an ***A*** usability grade. All prototypes scored above **80**, which indicates an above-average user experience [53].

Semi-structured interview

During the Post Test semi-structured interview, 10 out of 12 test participants said that they would like to use a system such as the one proposed in a gym environment. The 2 test participants that were not interested in such a system both implied that the system/concept was too complex and overbearing. Such a system/concept would be too distracting when trying to focus on working out and interacting with other people in the gym.

The test participants also provided feedback in regards to the prototypes' functionality and user interface. Examples of such feedback includes:

- The user flow of starting the Deadlift exercise was not straight-forward; there was more than one way of getting there, along with the a large amount of non-interactable/ functional interface elements and components made navigation confusing.
- In the case of the "Puck" prototype - when asked to reconnect to the system after having left the free weight training area to go get a drink of water; it was not clear *which* puck that the test participants needed to tap with their phone in order to reconnect.

  Test participants (correctly) observed that tapping whichever puck available to them in the gym (such as pucks found on gym machines) did not initiate an automatic free weight exercise tracking connection attempt. This was something that was brought up during the previous meeting with the stakeholders - but had not been addressed (see Section 5.1).

- Some form of animation of the illustrations when it came to instructing the user on how to connect to the system would help with understanding.

Table 5.4: The amount of test participants that ranked the different prototypes as their first, second or third choice.

Most preferred prototype			
Prototype	**1st place**	**2nd place**	**3rd place**
Puck	**5**	**4**	**0**
Waypoint	**4**	**1**	**2**
Gesture	**3**	**1**	**3**

Additionally, the test participants were asked which of the three prototypes they most preferred. The results and ranking of prototypes is shown in Table 5.4. The "Puck" prototype had the most 1st and 2nd place rankings - with the "Gesture" placing last due to its lowest 1st place and highest 3rd place rankings.

Chapter 6

Hi-Fi prototype

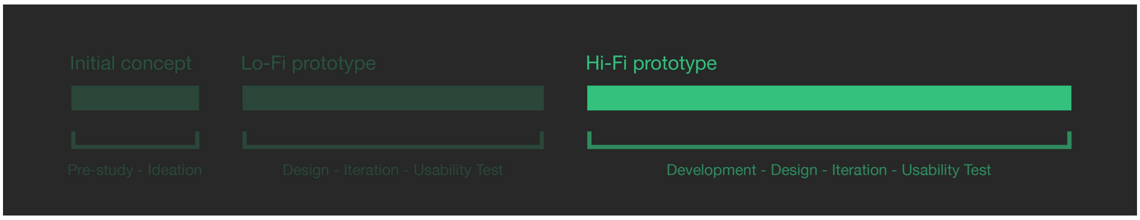


Figure 6.1: An overview of the current active phase of the project.

The goal of the last phase of the project (and third cycle of the HCD process) was to create a high fidelity (Hi-Fi) prototype that implements the concepts, functions and features proposed and tested in the two earlier phases (the initial concept- and Lo-Fi prototype phase). This prototype would then also be evaluated through a usability test with a larger number of test participants compared to the Lo-Fi usability test.

As previously stated - it was important for the stakeholders of the project that the Hi-Fi prototype was fully-functional (real-world) with no simulated parts. Therefore, a significant amount of development time was needed - which meant that this phase made up around 50% of the entire length of the three phases (see Figure 6.1).

6.1 Narrowing the project's scope

The amount of features and functionalities of the Hi-Fi prototype that required implementation had been deemed too large by both project supervisors. As such, the scope of the project needed to be narrowed.

What this practically meant, was that one of the three Lo-Fi prototypes had to be discarded - along with features that were not relevant to answering the research questions and/or sub-goals of the project.

As mentioned previously, this meant that a meeting with the project's stakeholders was set up in order to both present the results of the Lo-Fi prototype usability test, but also having a discussion on how to move forward in regards to the project scope.

6.1.1 Lo-Fi prototype presentation and discussion

The presentation given to the stakeholders of Advagym and Sony's R&D Center Lund Laboratory, contained a demo of the Lo-Fi prototypes used in the Lo-Fi usability test - along with the results thereof. From the results detailed in Section 5.3.6 - it was not directly apparent which of the three prototypes held the most promise in terms of usability/performance/user preference. Due to the potential amount of implementation required by the R&D-team - it was recommended that the "Gesture" prototype (Lo-Fi prototype 1) should be discarded moving forward.

The stakeholders agreed with the recommendation and gave the go-ahead to implement the other two proposals as Hi-Fi prototypes: the "Waypoint" and "Puck" prototypes (Lo-Fi prototype 2 and 3, respectively). The Advagym stakeholders were specifically keen on keeping the "Puck" prototype since that connection method closely resembled the connection methods their users were already accustomed to when using their system.

It was also decided that one of the R&D-team's engineers would assist in writing the code and scripts needed to implement a waypoint, and the filtration of the MQTT messages sent by the CTS broker located in the Sony gym at the Lund offices.

All additional feedback (and discussions/talking-points) were recorded for use when iterating further upon the prototypes.

6.1.2 Discarded features and functionality

With the feedback gained from the Lo-Fi prototype presentation (as mentioned above); the "Gesture" prototype was discarded (Lo-Fi prototype 1).

The following additional features and functionality would also be discarded, along with the motivation(s) for doing so:

- **Programs**: the sub-goals of the project stated that the prototype application be used for evaluation and further testing of the *CTS* and the *automatic free weight exercise tracking solution*. The "programs" feature/part of the prototype was therefore deemed non-essential, as it was only *one* of the *two* user flows used for starting exercises.

- **Exercise search and filtration**: since there were only five exercises that the CTS was able to detect and track - implementing a possibly complex search and filtration solution was not necessary.
- **Free weight selection screen**: due to the above point of having just five exercises and no need for search/filtration, having two separate screens when selecting exercises was also not needed.
- **Non-functional interface components**: icons, buttons, and any other type of interface component that did not have actual real-world functionality would be discarded - since not a single part of the prototype application would be simulated.

6.1.3 User interface re-design

After further discussions with Advagym's supervisor and the team's engineers, it became apparent that the code being written for the implementation of the Hi-Fi prototypes, would *not* be re-used if Advagym were to adopt the CTS for a automatic free weight exercise tracking solution. This mostly had to do with the complexity of their existing code base, along with the legal/licensing issues presented when using external libraries in the upcoming Hi-Fi prototypes' implementation.

Therefore, it was decided that the Hi-Fi prototypes would *not* need to closely resemble/mimic Advagym's current smartphone applications on iOS or Android. This would in turn allow for simplification of the user interface layout, components of the prototypes. It would also allow more freedom to experiment with different screens and user flows.

6.2 Development and implementation

It was decided that the Hi-Fi prototype smartphone application would be built from scratch. The reasoning behind that decision included what was previously mentioned in Section 6.1.3: none of the code being written for the implementation of the application would be re-used by Advagym. Another part of the reasoning had to do with the large amount of time that would need to be spent on understanding and being able to build Advagym's applications using their current code base.

The app was based on the layouts and design decisions derived from the Lo-Fi prototypes and their usability test. Several software tools and devices were used to support the development:

- The Android version of the application ran on a *Google Pixel 2 XL* smartphone.
- The iOS version of the application ran on an *iPhone 13 Pro* smartphone along with a simulated iPhone 8 smartphone, running on a macOS laptop computer.
- Code was written in Microsoft's *Visual Studio Code*.

The entire application was written in the *TypeScript* programming language which is a syntactical superset of the JavaScript programming language, that adds optional static typing to the language [54]. In order for the application to build and run on both the iOS and Android operating system (per the sub-goals of the project), the cross-platform framework *React Native* was used.

Using React Native also allowed for a shared code base with only a few custom solutions needed for the different platforms (further detailed in Section 2.5). Due to the aforementioned limited amount of overall development time for the project, a few open source libraries were also used when implementing/supporting many of the technologies needed for the fully-functional features and functionalities of the app.

The open source libraries and packages were managed and installed through *npm*, a package manager and software registry [55]. The different implementations and parts of the application required for the creation of the Hi-Fi prototypes are outlined below.

6.2.1 User interface

For the user interface of the application, components such as buttons, lists, switches and navigation bars were mostly imported from React Native's built-in component library. However, almost all of these components had to be re-written or modified in ways to make the look-and-feel consistent across both iOS and Android, since React Native defaults to the platforms' own version of the components, whilst maintaining the functionality. Some components such as buttons and *modals* (pop-ups) had to be built from scratch since there were no comparable methods/components available from React Native.

Navigation around the application was implemented through React Navigation, which also allowed for animation affordances. In order to allow for consistent navigational behavior across platforms - gestures for "going back" had to be overridden since iOS and Android allows for different interactions.

Illustrations and icons used throughout the interface were created in Sketch and then exported as Scalable Vector Graphics (SVG) files. This allowed for scaling without detail-loss across different resolutions/screen sizes of smartphone devices. Debugging and running React Native applications allows for multiple smartphone devices (and/or simulators) to be updated in real-time, when making code changes. This feature is known as "Hot Reloading" [56]. This allowed for very quick iteration of the user interface; making sure that any changes made to the interface looked acceptable across all devices/platforms/screen sizes/resolutions that were available for testing.

6.2.2 State handling and application settings

In order for the application to understand what information to display to the user, or what feedback to provide in response to system errors or actions taken by the user - the application would need to handle the following *states*:

- **LoggedIn**: is the user logged in or not?
- **UserID**: what is the ID of the logged in user?
- **UserType**: is the user an admin/tester?

- **Connection**: this state consists of the following child-states:
  - **Connecting**: is the user attempting a connection?
  - **Connected**: is the user connected to the CTS or not?
- **Error**: an error of some kind occurred.

The state handling was implemented through the use of React Redux [57]. Redux is a global state management tool. It allows for components to access the global updated state through the use of "Hooks" [57]. This was used to update the appearance or functionality of components or layouts of different parts of the application. The states were stored either as *boolean*, *string* or *number* types.

There was also data that needed to be stored in memory somehow, that the application could both read and write to. Redux allows for a "store" which can save and handle data of different kinds. This was used extensively throughout the application. Some of the data stored were as follows:

- **skeletonID**: the user's current "Skeleton ID" according to the CTS (saved as a string).
- **latestMessage**: the latest MQTT-message received from the CTS (saved as a string).
- **repNumber**: the current number of repetitions in the set (saved as a number).
- **exercises**: an array of all exercises performed by the user, and the accompanying data of each exercise (its number of sets, repetitions, weight lifted etc. - was saved as an array containing strings and numbers).

The application also needed to be able to store user settings that would persist through application shutdowns and startups. Redux also allowed for this through the use of its "persist" feature [57]. Such settings and along with their data type, included:

- **audioFeedback**: if the user wanted auditory feedback or not (saved as a boolean).
- **connectionMode**: the connection method ("puck" or "waypoint") used with the CTS (saved as a string).
- **brokerAddress**: the IP address of the CTS broker (saved as a string).
- **brokerPort**: the port of the IP address that one can listen to for MQTT messages (saved as a string).
- **debugMode**: display debug information or not (saved as a boolean).

In order for states in the application to be updated/changed, the application needs to be connected to the CTS.

6.2.3 CTS connection

In order for the user to connect and interact with the CTS - the prototype smartphone application needs to be able to:

- Connect and listen to the CTS broker's modified topic and parse its MQTT messages.
- Know *when* to listen for certain MQTT message types.
- Know what Skeleton ID(s) are within the "waypoint" boundaries, in the tracked area.

CTS broker connection, modified topic and MQTT message parsing

The CTS communicates through MQTT messages that are broadcast from a broker on a certain IP address and port, on a Wi-Fi network. That meant that an MQTT "client" service had to be implemented inside of the application that constantly listened for messages on that IP address and port.

The amount of messages that needed to be received and parsed by the React Native application proved to be too much for a smartphone device to computationally handle. MQTT clients receive MQTT messages by subscribing to a topic that the broker has created. As such, a new topic needed to be created that only sent messages that were *relevant* to the automatic tracking solution. This was done by filtering the MQTT messages through the use of a script written in Python (a programming language). The filtration provided that only the three following *message types* were sent:

- **activity**: the exercise performed, and what Skeleton ID performed it.
- **exited**: the Skeleton ID(s) that were no longer present in the tracked area.
- **closest**: the Skeleton ID(s) that were within a certain distance to the "waypoint".

Each message was received by the application in the form of a JSON string that was then parsed into a JSON object with data contained inside.

Depending on the message type - states and/or data are changed in the application. For instance, when the CTS lost track of the connected user - the following would occur:

1. A *exited* message is received by the application, whilst it is in the *Connected* state.
2. The *latestMessage* is updated to hold the message's parsed JSON object.
3. The application checks if the latestMessage's array of *skeletonIDs* contains the *skeletonID* string saved in the application.
4. The *skeletonID* is present in the array.
5. The application updates its *Error* state to "trackingLost".
6. The application updates the *Connected* state's boolean to "false".
7. The application's user interface updates its appearance through the use of hooks that were listening to the *Connected* and *Error* states.
8. A modal is shown to the user, telling them that they have been disconnected from the CTS (due to tracking being lost).

Remote development, debugging and work on the project proved difficult - since you would have no access to the "real" CTS located at the Lund office. To solve this - a simplified, "simulated" CTS was created. The system consisted of a *Mosquitto* broker service and a simple shell script that listened for specific keystrokes, running on a laptop computer [58].

In order to use the "simulated" CTS - the application's *brokerAddress* and *brokerPort* settings would be changed to match that of the Mosquitto broker. Then - depending on the key entered - an MQTT message was sent to the Mosquitto broker. The messages sent, were identical to that of the messages sent by the "real" CTS.

Knowing when to listen to certain MQTT messages

To initiate a connection attempt to the CTS, the smartphone application needed to leverage the two proposed connection methods: the "puck" and the "waypoint".

In the case of the "puck" connection method, the smartphone application needed to interact with Advagym's IoT "puck" device through NFC or BLE. Advagym's current application for Android uses NFC, whilst their iOS variant of the application uses BLE. The same paradigm was therefore also used for the prototype application.

If the NFC service is initiated on the smartphone application, it automatically starts listening for other NFC devices. If a user taps their smartphone to a puck, the NFC tag in the puck would be picked up and identified by the service. A quick cross-reference is then made between the transmitted identifiers of the NFC device and that of Advagym puck identifiers. If they match (i.e. a puck is detected), the application updates its connection state to *Connecting*.

If the smartphone application is running on an iOS device - an "iBeacon" method is used to communicate with the puck over BLE [59]. When the iBeacon service is initiated on the smartphone application, it will start to listen to other "beacons" in its vicinity. The Advagym puck device does not constantly broadcast as a beacon. In order to make the puck broadcast over BLE, the proximity sensor on the front of the puck needs to be covered. Since tapping the puck with a smartphone fulfills that requirement, you get the same user interaction as with the NFC implementation. As with the NFC implementation - when the iBeacon service picks up a beacon - that beacon's identifiers are cross-referenced with Advagym's puck identifiers. If they match, the application - as with the NFC implementation - updates its connection state to *Connecting*.

In the case of the "waypoint" connection method, only a simple "CONNECT" button press is needed for the smartphone application to enter the above mentioned *Connecting* state.

When the smartphone application is in the *Connecting* state, it will start to listen for the *closest* MQTT message types. In all other states, messages of this type are discarded and not acted upon.

The "waypoint", its boundaries and Skeleton ID(s)

This was the key part of the solution for connecting and using the CTS. In order for the CTS broker to send out the *closest* MQTT message type, it needs to know/do two things:

- The X, Y and Z coordinates of the "waypoint".

- Detect a skeleton (tracked human) within a certain distance of the "waypoint" (i.e. within the "waypoint boundary").

The "waypoint" location could be anywhere, as long as it was inside of the tracked area of the CTS. In order to define a waypoint, the following was done:

1. Make sure the tracked area has no other humans in it.
2. Stand at the exact location of where you would like the waypoint to be.
3. Record the X, Y and Z coordinates of the Skeleton IDs present in the scene reported by the CTS.
4. The recorded X, Y and Z coordinates of the single, stationary "Skeleton" is your "waypoint" location.

In order to determine that a skeleton is within the "waypoint" boundary, the formula for calculating the distance, d between two points $P_1 = (x_1, y_1, z_1)$ and $P_2 = (x_2, y_2, z_2)$ in 3D-space (xyz-space) was used (see Equation 6.1):

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{6.1}$$

If the distance d between the waypoint and a skeleton was calculated as less than $0.5m$, a *closest* message was sent out by the CTS broker containing the Skeleton ID of the tracked/detected skeleton.

6.2.4 Feedback

Depending on what states or what features the user was interacting with, feedback of different kinds needed to be implemented within the application.

A lot of the feedback utilized, had to do with the on-screen user interface and its iconography, animations and changing of colors of the components, etc.. However, since the users of the automatic tracking system might not always have their smartphone in direct view when performing exercises, more modes of feedback needed to be supported.

Haptic feedback in the form of vibrations from the vibration motors of the smartphones used to run the application was implemented when certain things occurred within the app. Such things would include when an error occurred, or when the user was disconnected from the CTS - or when the user successfully connected to the CTS. The strength and duration of the vibrations could be controlled through the use of built-in functions within React Native - which could signal the *type* of feedback to the user.

Another form of feedback that would be used for the Hi-Fi prototypes was that of *auditory feedback*. This was implemented through an open source "text-to-speech" (TTS) library: *react-native-tts* [60]. The functionality provided by the library was used to let the application "speak" to the user, informing them of things happening on-screen. This was done through the input of a simple string into a specific "speak"-function of the imported TTS package.

6.2.5 APIs and logging

One of the most important sub-goals of the project was to allow for evaluation of the CTS. In order for Advagym and Sony's R&D Laboratory Lund to do that, the application needed to offer different functionalities and features - depending on the user type.

What this meant in practice, was that certain settings and features of the application were closed-off to "normal" users - whereas "admin" users had the ability to fully debug the CTS using these aforementioned settings and features. These settings included things such as setting the broker address and port, or changing the connection method of the application.

In order to do this, it was decided that the application would utilize Advagym accounts. When logging into an Advagym account, the Advagym UserID would then inform the application of the active user type; resulting in the app changing its *UserID* and *UserType* states. The login procedure was implemented by interacting with Advagym's login API through *Axios* - a client library that serves to create HTTP requests [61].

Using Axios, a *POST* request was sent to the Advagym's servers with the login information provided by the user (in this case, an email and password). After having received a response signaling a successful login, the user was logged into the app with the correct *UserID* and *UserType* set.

It was also decided that the application would support registering completed user workouts (with its exercises, sets, reps, weights and duration), with the logged in Advagym account. This was also done through Axios, but this time through a *PUT* request. The PUT request's data payload needed to be a specifically formatted JSON string in order for their system to register the workout correctly. Therefore, a formatter was written, that took workouts and exercises stored in the application as the input. The output was the aforementioned specifically formatted JSON string, which was then sent off.

Lastly, it was important that log data could be collected with the help of the application. If a user encountered incorrect behavior or bugs relating to the CTS - a specific timestamp of the incident along with relevant information as to what happened, needed to be sent to the team's engineers. It was proposed that this could be done through the use of *Slack Webhooks* [62]. This implied that logging messages were posted to a specific workspace channel within *Slack*, a team messaging software solution [63]. The stakeholders accepted the proposal and an implementation was created through the use of yet another Axios POST request.

6.2.6 Project management and versioning

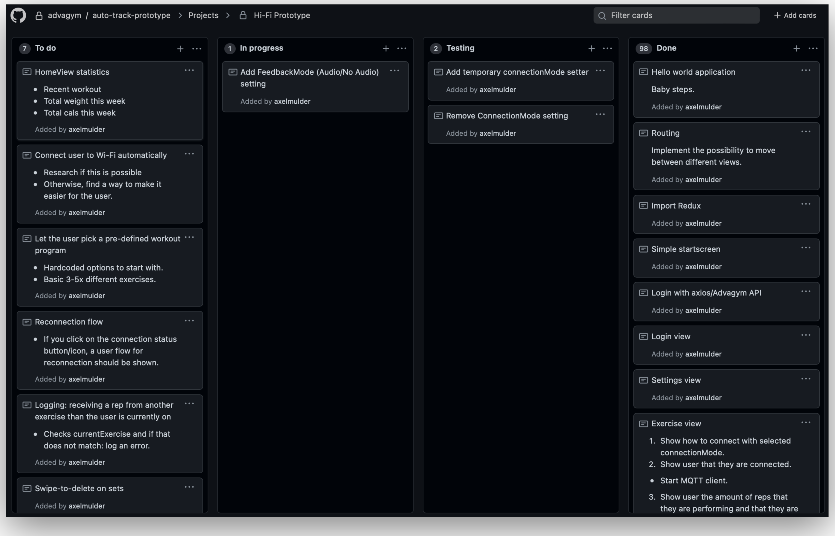


Figure 6.2: An overview of GitHub Issues' kanban board, where one can see cards with tasks that are in different phases of completion.

The development of the smartphone application was managed through *GitHub*, and one of its tools, called *Issues* [64, 65]. The tool was used to create a *Kanban Board* that was used to plan out and visualize the current progress of the features, functionalities and bugs that needed to be taken care of during the project (see Figure 6.2). The kanban board was also shared with the project supervisors in order for them to monitor the project's progress in real-time.

The versioning (iteration) of the application was handled through *Git*, which is an open source distributed version control system [66]. The repository containing the application's code base and assets is hosted on a private repository within Advagym's GitHub organization.

6.3 Prototyping

After the development and implementation concluded, a fully functional Hi-Fi prototype smartphone application had been produced. Through the narrowing of the scope of the project, some screens were discarded - but new ones were also added to support features such as login:

- Login screen
- Home screen
- Settings screen
- Exercise Selection screen
- Exercise screen
- Current Exercise screen
- Workout screen

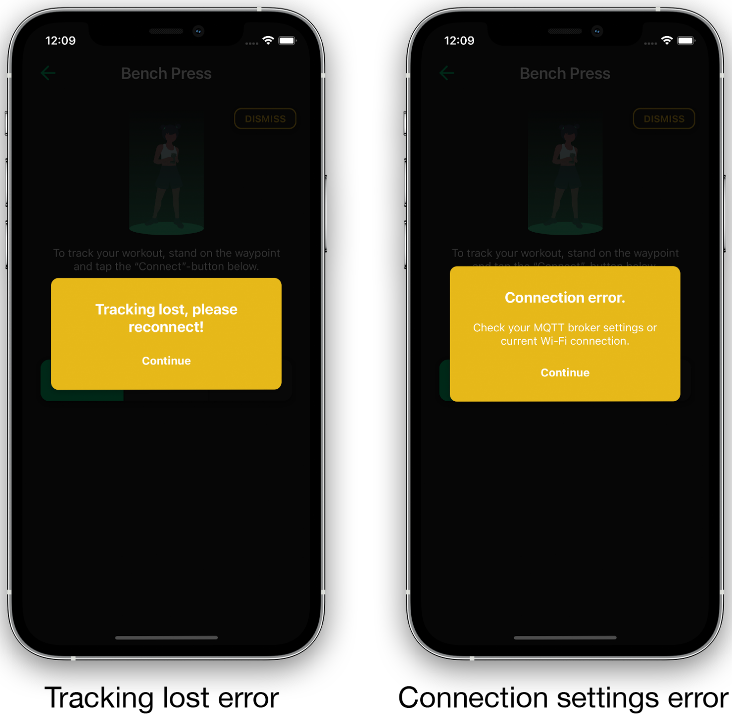


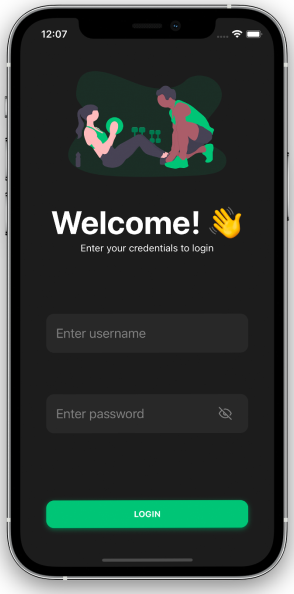
Figure 6.3: The Hi-Fi prototype showing an error where the system has lost track of the user, along with an error indicating to the user that they need to check their broker IP/Wi-Fi connection settings.

There are notable differences between the Lo-Fi prototype screens and that of the Hi-Fi prototype screens. One of the largest changes came in the form of an application-wide (i.e.

is shown irrespective of what screen the user is on), full-screen modal (pop-up) that was shown to the user if a system error occurred (see Figure 6.3 for examples). Additionally, if the system lost track of the user - and the user had enabled the auditory feedback setting - the smartphone running the application would use its built-in speakers to speak to the user and say: "Connection lost, please reconnect".

Comparisons with the earlier Lo-Fi prototypes, along with the iterations and changes made are detailed below - followed by a presentation of the two Hi-Fi prototypes used in the Hi-Fi usability test:

Login screen



Login screen

Figure 6.4: The login screen from the Hi-Fi prototype.

In order for the user to use the application, they first need to login with an Advagym account (see Figure 6.4). This was done in order to support permissions in regards to management and debugging of the CTS. After a successful login, the user is navigated to the Home screen:

Home screen

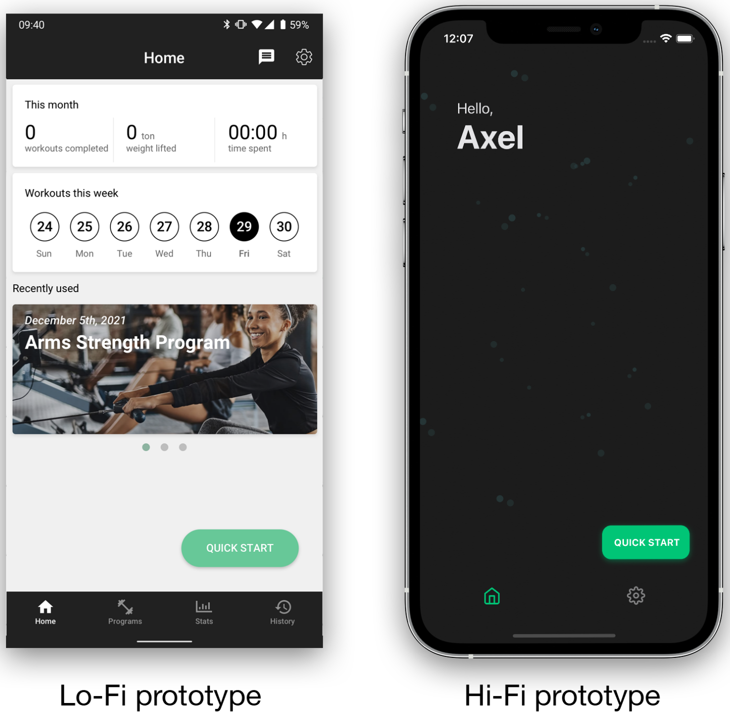


Figure 6.5: Comparison between the Lo-Fi prototype's and Hi-Fi prototype's Home screen.

The Home screen of the Hi-Fi prototype was simplified and only contained two possible interactions: either tapping/clicking on the Settings cog-icon on the bottom navigation bar, or clicking the "QUICK START" button (see Figure 6.5).

As previously mentioned, the scope of the project was narrowed which meant that the user statistics shown in the Lo-Fi prototype's Home screen would not be implemented due to the effort required for doing so. Instead, the user of the application is met with a simple greeting (the name used in the greeting is taken from the user's Advagym account that was used to login).

Settings screen

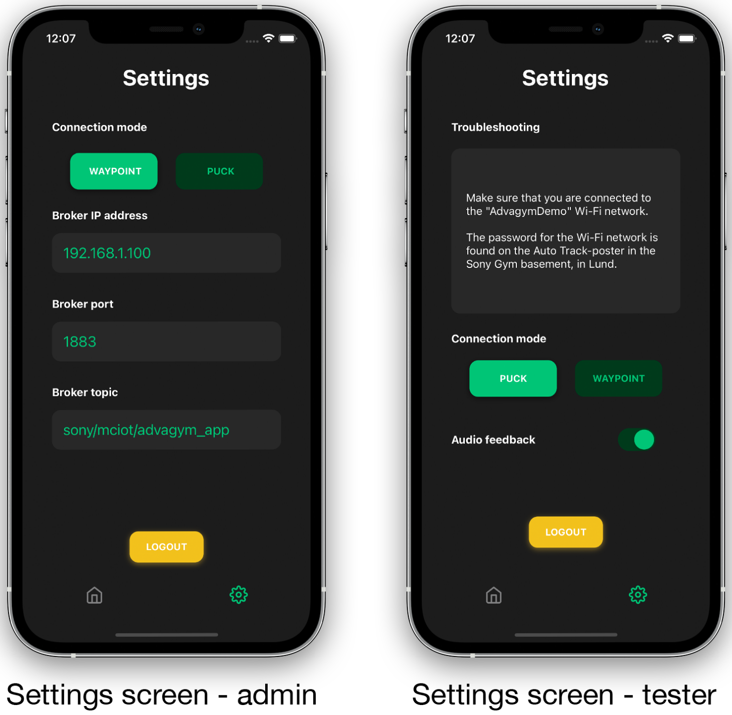


Figure 6.6: The Settings screen shown to a user with the *user type* "admin" followed by the Settings screen shown to a user with a "tester" user type.

The user navigates to the Settings screen by tapping/clicking on the cog-icon found in the bottom navigation bar on the Home screen. Depending on the *user type*, different application settings and functions are available to the user (see Figure 6.6). If a "normal" (or "tester") user type is logged in, troubleshooting instructions on how to connect and use the application are shown at the top of the screen.

Tapping/clicking on the house-icon in the bottom navigation bar returns the user to the Home screen.

Exercise Selection screen

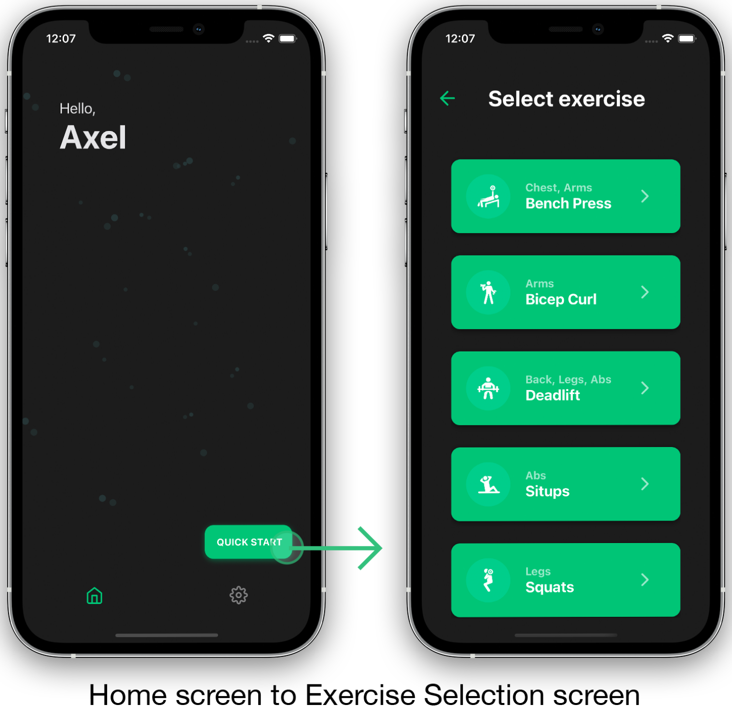


Figure 6.7: Result of tapping/clicking on the "QUICK START" button on the Home Screen.

In order for the user to navigate to the Exercise Selection screen, the user simply taps/clicks on the "QUICK START" button (see Figure 6.7). Since the "Programs" feature/screen of the Lo-Fi prototype was discarded - this is the only way for the user to start using the application (i.e. starting a workout).

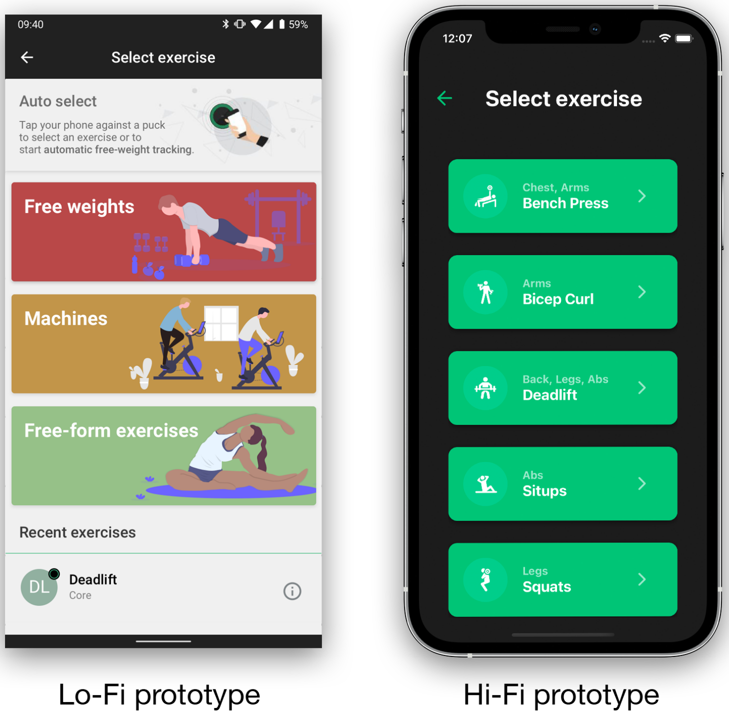


Figure 6.8: Comparison between the Lo-Fi prototype's and Hi-Fi prototype's Exercise Selection screen.

The Hi-Fi prototype is heavily simplified compared to the Lo-Fi prototype (see Figure 6.8). As mentioned previously, since the CTS only supports five exercises, there was no need for a complex UI with search and filtering functionality. Iconography for each individual exercise was created in order to aid with the discoverability-factor of the interface.

Exercise screen

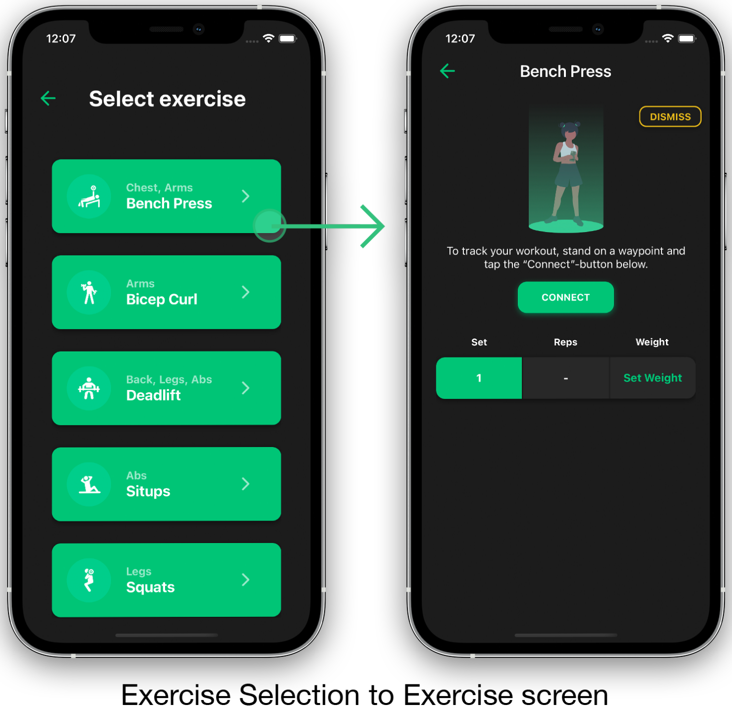
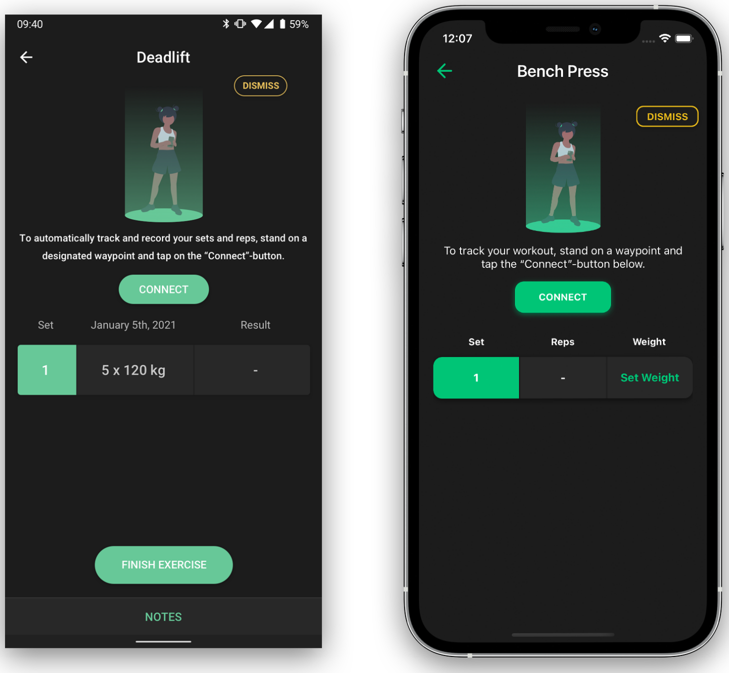


Figure 6.9: Result of tapping/clicking on the Bench Press exercise on the Exercise Selection Screen.

Navigating to the Exercise screen from the Exercise Selection screen is done by tapping/clicking on any of the available green exercise "cards" (see Figure 6.9).



Lo-Fi prototype Hi-Fi prototype

Figure 6.10: Comparison between the Lo-Fi prototype's and Hi-Fi prototype's Exercise screen.

As can be seen from Figure 6.10, there are not many differences between the two prototypes; the "FINISH EXERCISE" button from the Lo-Fi prototype has been moved, and is now only seen when the application is in the "connected" state. The "Notes" tab found in the bottom of the Lo-Fi prototype screen was also discarded since it wasn't relevant to the sub-goals and research questions of the project.

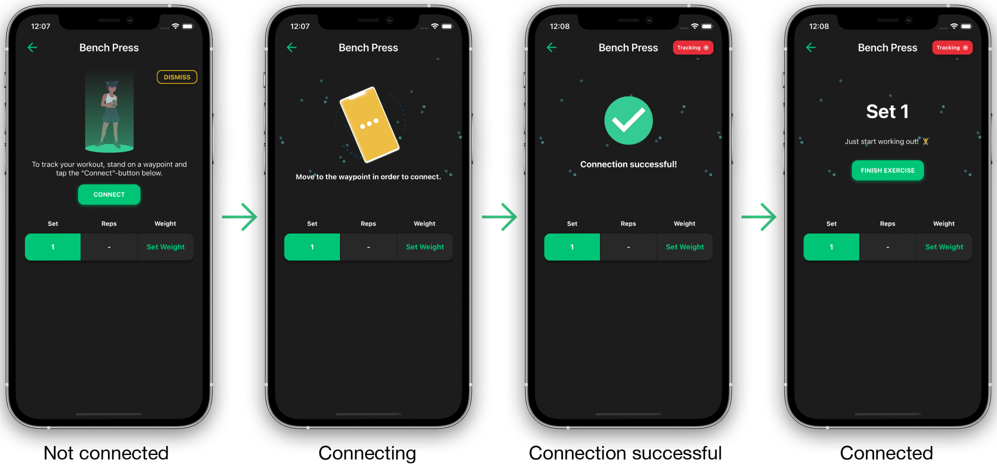


Figure 6.11: The Hi-Fi prototype's four connection states shown in the Exercise screen.

Similarly to the Lo-Fi prototype, the different connection states of the application makes the top of the Exercise screen change its appearance (see Figure 6.11). Additional feedback had been added in the form of a vibration if the user successfully connected to the CTS. Also, if the user had the auditory feedback setting enabled, a human voice from the smartphone's speakers would say "Connection successful".

An additional affordance was made in the connected state of the Exercise screen, in order to better guide the user: a short instruction informed the user that the system was ready to go and that the user could go ahead and start doing their exercises (see the last screen in Figure 6.11).

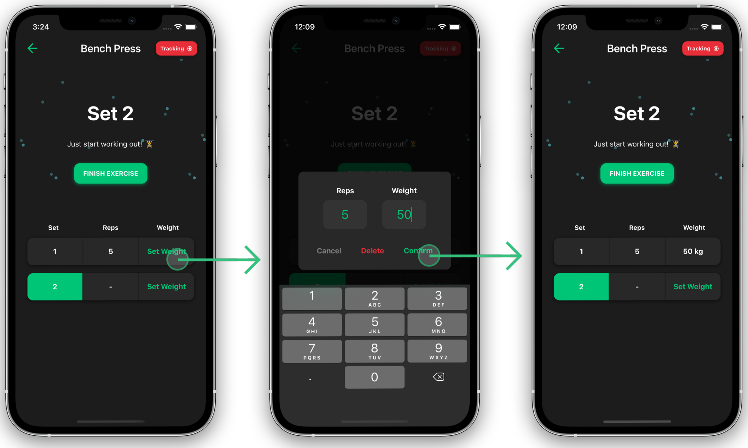


Figure 6.12: Result of tapping/clicking on a set and editing the weight amount on the Exercise screen.

A new feature of the Hi-Fi prototype was the ability for the user to edit and remove sets. This was done by tapping/clicking on the set in the set list found in the lower part of the screen (see Figure 6.12). Since the R&D stakeholders wanted to log the errors encountered by users whilst using the CTS; if the user edited the amount of repetitions in an automatically tracked set, a log was saved and sent through the Slack webhook. The log contained the start- and stop time of the exercise performed, Skeleton ID and what the user changed the repetition from and to.

Manually disconnecting from the CTS or changing into a "manual" workout mode where the user can create sets on their own, was implemented in the same exact way as the Lo-Fi prototype (see Figures 5.9 and 5.11).

Current Exercise screen

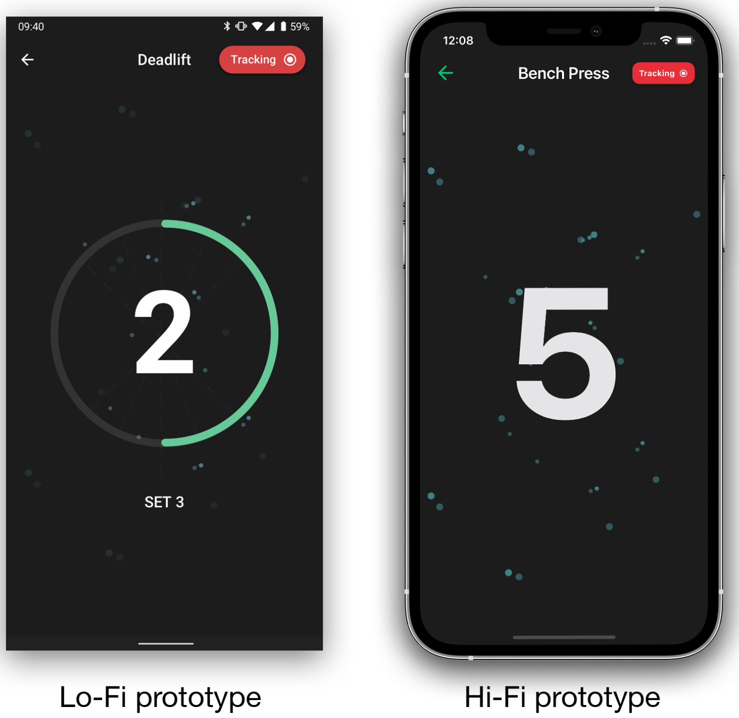


Figure 6.13: Comparison between the Lo-Fi prototype's and Hi-Fi prototype's Current Exercise screen.

The Current Exercise Screen of the Hi-Fi prototype did not differ very much from the Lo-Fi prototype's equivalent screen (see Figure 6.13). The planned green ring around the rep counter that would be "filled up" clockwise as the user would do more repetitions was removed since the "Programs" feature of the prototype had been discarded. The text with the current set number was also removed since it was deemed unnecessary information on that particular screen.

This screen was automatically navigated to when the application receives an *"activity"* MQTT message from the CTS broker that contains the user's current *Skeleton ID* as well as the name of the detected exercise, matching the current exercise's name which is shown at the top of the screen (see Figure 6.13).

Whenever a new *activity* MQTT message with the requirements stated above was received, the repetition counter on the screen would be incremented by one. If no *activity* MQTT messages had been received for six seconds, the application considers the current set completed, and the user is returned to the Exercise Screen which now shows an updated sets list.

If the application received an *activity* MQTT message with a Skeleton ID that does *not* match the user, it was discarded. The same thing applies to if the application receives an activity MQTT message that contains an exercise that does *not* match the user's "current exercise" i.e. matching the exercise screen that the user is currently on.

Workout screen

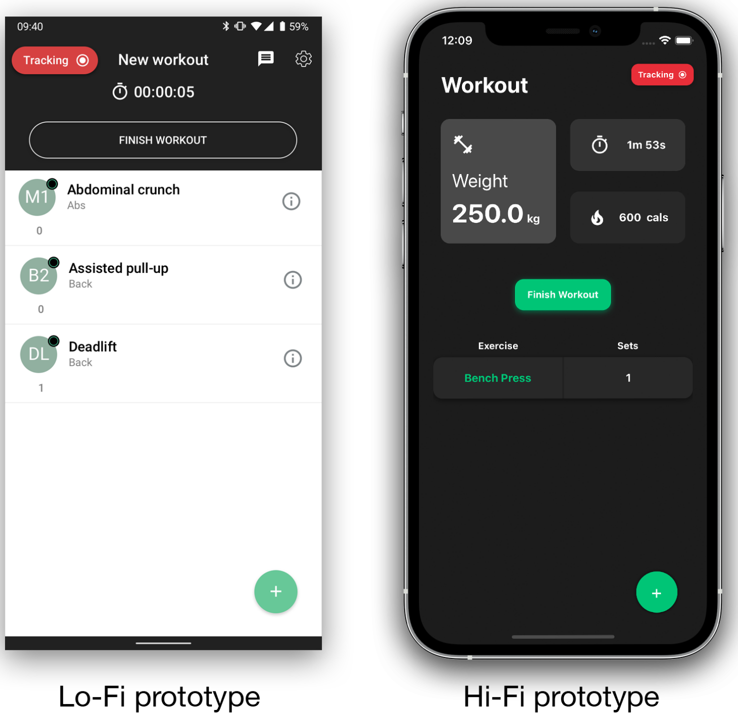


Figure 6.14: Comparison between the Lo-Fi prototype's and Hi-Fi prototype's Workout screen.

Here we can see a substantial difference between the Lo-Fi and Hi-Fi prototype (see Figure 6.14). Statistics of the current workout was calculated and added to the top of the screen that updated whenever the user completed sets and exercises.

The same navigation and functionality of the Lo-Fi prototype's Workout screen also applied here:

- You could get to the Workout screen by clicking the "FINISH EXERCISE" button in the Exercise screen, or when navigating "back" by using the left-facing arrow at the top left of the screen after having completed at least one set.
- Tapping/clicking an exercise brought the user back to that exercise's Exercise screen.
- Tapping/clicking the FAB in the bottom-right corner brought the user to the Exercise Selection screen.
- Tapping/clicking the "FINISH WORKOUT" button brought the user back to the Home screen.

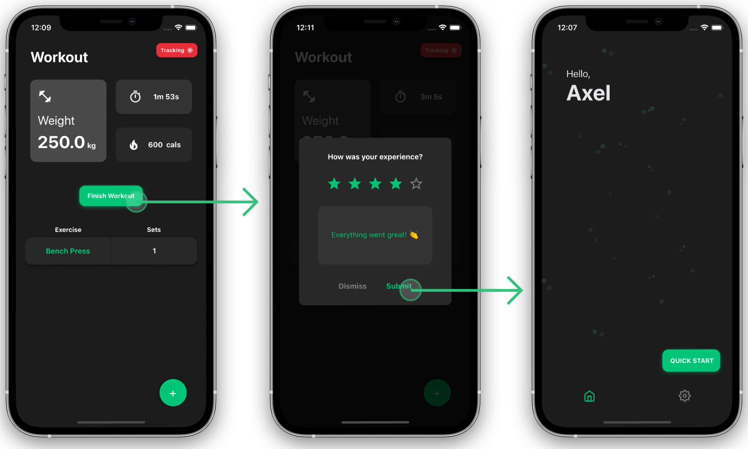


Figure 6.15: Result of clicking/tapping on the "FINISH WORKOUT" button on the Workout screen.

However, tapping the "FINISH WORKOUT" button did not bring the user back immediately back to the Home screen. First, a modal (pop-up) was shown, asking the user to rate the experience of using the automatic free weight exercise tracking solution - and leave any feedback they could think of (see Figure 6.15). This was then logged and sent to R&D team through a Slack Webhook.

Additionally - when a workout was finished by the user - the workout data itself with its exercises, sets and weights was also saved and uploaded to their logged in Advagym account.

6.3.1 Hi-Fi prototype 1 and 2

As with the Lo-Fi prototypes, the Hi-Fi prototypes were created to test and evaluate the proposed connection methods. Since the "Gesture" connection method and prototype had been discarded, only the "Puck" and "Waypoint" prototypes remained.

Hi-Fi prototypes 1 and 2 utilized the "Waypoint" connection method and used the same connection flow used in Lo-Fi prototype 2. The difference between Hi-Fi prototypes 1 and 2 had to do with the amount of feedback afforded to the user:

Hi-Fi prototype 1 had *no* additional feedback beyond what was displayed on the screen of the user's smartphone. This meant that when the user was doing exercises, or moving around in the tracked area; unless they were looking directly at the screen, they would perhaps be unaware of what was currently happening from the CTS's perspective.

Hi-Fi prototype 2 however, had additional feedback in the form of auditory feedback. The auditory feedback as mentioned throughout the Section 6.2.4 and 6.3, was implemented through a TTS solution. Whenever the user needed to be aware of what the CTS was "seeing", the sound of a human voice played - informing the user:

- When the connection to the CTS was successful.
- When the system lost track of the user.
- When the system detected an exercise.
- When the system considered the current set of exercises completed.

6.3.2 Hi-Fi prototype 3 and 4

Hi-Fi prototypes 3 and 4 were created in order to evaluate the "Puck" connection method and used the same connection flow used in Lo-Fi prototype 3. As with Hi-Fi prototype 1 and 2, the difference between the Hi-Fi prototypes 3 and 4 had to do with the amount of feedback afforded to the user:

Hi-Fi prototype 3 - as with Hi-Fi prototype 1 - did not have any additional feedback beyond what was displayed on the screen of the user's smartphone.

Hi-Fi prototype 4 had the same auditory feedback as Hi-Fi prototype 2. Additionally, it was afforded haptic feedback. A vibration was used when the user's smartphone was tapped to the Advagym "puck", indicating that a successful connection with the puck had been registered.

6.4 Hi-Fi usability test

After the completion of the development and implementation of the Hi-Fi prototypes, a usability test was conducted.

The goal of the Hi-Fi usability test was to use its results to answer the research questions posed in this thesis work, along with validating that the sub-goals drawn up by Advagym and Sony's R&D Laboratory Lund had been met.

As with the Lo-Fi usability test, a test plan was created that detailed the methodology used, the data being collected, what resources were required and how the test participants were recruited and briefed. This test plan was also reviewed and approved by both project supervisors.

6.4.1 Test design

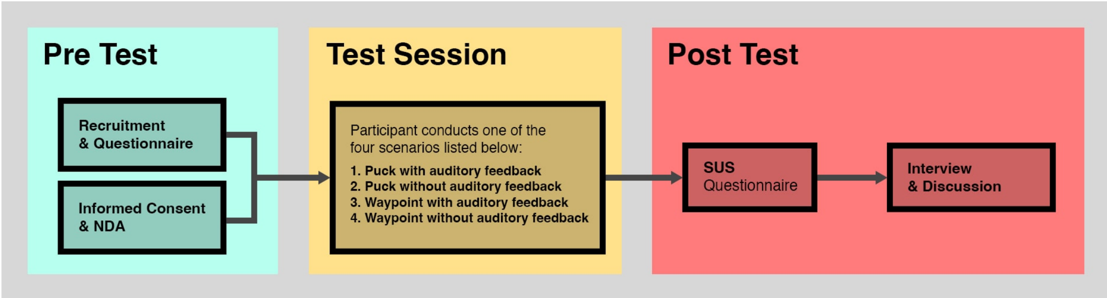


Figure 6.16: The different parts and workflow of the Hi-Fi usability test.

The Hi-Fi usability test was carried out in three parts: "Pre Test", "Test Session" and "Post Test". The different parts were carried out in a specific order with the Pre Test being the first, followed by a Test Session and lastly a Post Test (see Figure 6.16).

Pre Test

This part/phase of the test consisted of the recruitment, and initial contact with the test participants. It was structured and carried out in the same way as the Lo-Fi usability test's Pre Test (see Section 5.3.1).

However, since the test participants were going to be filmed with a video camera during the test session, this needed to be clearly communicated during the informed consent and NDA signing.

Test Session

By the time of the test sessions for the Hi-Fi prototype, the COVID-19 pandemic regulations and restrictions had been removed from Sweden. Therefore, all test participants were able to conduct the usability test sessions in-person.

Figure 6.17: A picture of the Sony offices' gym, showing the tracked area and cameras of the CTS - along with the location of the "way-point" and "puck".

The test sessions took place in Sony's office gym located in Lund (see Figure 6.17). The gym itself was equipped with fitness benches, a diverse range of gym machines, along with free weights in the form of barbells with weights, dumbbells, etc.. Throughout the gym and along the ceiling, small cameras were placed that together created the tracked area of the CTS.

Figure 6.18: A picture of the "waypoint" and "puck" used during the test sessions for the Hi-Fi prototypes.

In the middle of the room, adhered to a larger gym machine assembly, was the "puck" used for Hi-Fi prototypes 3 and 4 (see Figure 6.18 and 6.17). On the floor beneath the puck, a large circle was outlined with the help of tape. The circle functioned as the "waypoint" used for Hi-Fi prototypes 1 and 2 (see Figure 6.18 and 6.17).

At the start of the test session, the test participants were given a short introduction and background as to why they were there. The participants were then given the choice of either an Android or iOS smartphone device to use during the test. The test participant was encouraged to use the device and operating system that they were most accustomed to - making it one less "step" in order to understand and use the application correctly.

When the participant was ready, they were be asked to perform specific tasks when using one of the Hi-Fi prototypes:

- Hi-Fi prototype 1: Waypoint, that did *not* have auditory feedback.
- Hi-Fi prototype 2: Waypoint, that had auditory feedback.
- Hi-Fi prototype 3: Puck, that did *not* have auditory or haptic feedback.
- Hi-Fi prototype 4: Puck, that had auditory and haptic feedback.

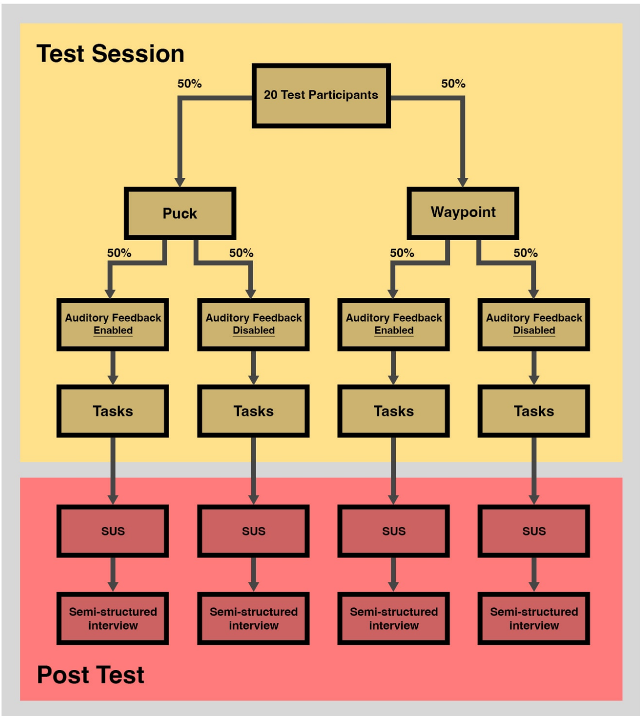


Figure 6.19: The Test Session's participant distribution and overall structure; consisting of its four test scenarios - which then leads into the Post Test.

Since the test participant would only use *one* of the Hi-Fi prototypes - that meant that the four Hi-Fi prototypes functioned as the different *test scenarios* (see Figure 6.19).

The tasks that the participants were asked to complete, were the following (regardless of what scenario they have been assigned):

Task 1 : Carry out two sets of Squats, five repetitions each, with the help of the application and the automatic tracking feature.

Subtask 1.1 : Connect to the CTS.

Subtask 1.2 : Understand that two sets, with five repetitions each, have been recorded correctly *or* incorrectly, by the application.

Task 2 : Go over to a table outside of the tracked area (main exercise area) - to get some water - and then return to the exercise area.

Subtask 2.1 : Understand that the tracking has been lost.

Task 3 : Carry out one set, ten repetitions of Bicep Curls, with the help of the application and the automatic tracking feature.

Subtask 3.1 : Reconnect to the CTS.

Subtask 3.2 : Understand that one set with ten repetitions has been recorded correctly *or* incorrectly, by the application.

Task 4 : Carry out two sets, five repetitions of Bench Press - with the help of the application and the automatic tracking feature.

Subtask 4.1 : Reconnect to the CTS (if applicable).

Subtask 4.2 : Understand that two sets, with five repetitions each, have been recorded correctly *or* incorrectly, by the application.

Task 5 : Carry out a couple of Bicep Curls repetitions - but do it without the application automatically tracking the repetitions/exercise.

Task 6 : Carry out a couple of additional Bicep Curls repetitions - with the help of the application and the automatic tracking feature.

Task 7 : Finish/end the workout.

The subtasks associated with the tasks listed above were *not* communicated to the test participant, only the "main" tasks were (Task 1, Task 2, etc.).

Post Test

After all tasks had been completed, the test participants were asked to fill out a SUS questionnaire regarding that specific prototype (see Appendix B; all SUS questionnaires were identical to one another).

Lastly, a semi-structured interview was conducted, where the following questions were be asked:

1. Have you used any gym training application before? If so: which one(s)?
2. What are your thoughts regarding connecting and being tracked by the system?

3. Did you understand when you were - and when you were not - tracked by the system? Did you feel in control?
4. What are your thoughts regarding working out with the help of the system?
5. Did you think that the tracking system performed/worked as you had thought?
6. Do you have any improvement ideas or changes you would like to see with regards to the product/prototype?
7. Any other comments?

If there were any other questions that came to mind whilst observing the participant completing tasks during the test session - they would be asked in this portion of the test about it.

6.4.2 Data collection

As with the data collection seen during the Lo-Fi usability test, quantitative and qualitative (subjective) metrics and data were collected throughout the duration of the Hi-Fi usability test.

The type of data collected from the Pre Test was identical to that of the Lo-Fi usability test.

During the test session, the participants' interactions with the prototypes were video (and audio) recorded. The resulting metrics with their associated fail/pass criteria was recorded and included the following:

- **Success**: task was completed with no assistance required within a reasonable amount of time depending on the task.
- **Non-critical error**: task was completed with minimal assistance, such as a minor hint and/or restating the task objective.
- **Critical error**: task was not completed or was completed with major assistance.
- **Time on task**: the amount of time each individual task took for the test participant to complete.

Just as with the Lo-Fi usability test, the results of the SUS questionnaires filled out by the test participants were used to create SUS scores and a mean SUS score for each prototype. In turn, these mean SUS scores were used to provide letter grades for the different prototypes.

Lastly, subjective data from the answers from the participants answers to the Post Test interviews was collected - to determine the system's and prototype's understandability, and if the test participants would like to use a solution such as the one proposed - in a gym setting.

6.4.3 Resources

The Pre Test was conducted over the internet. However, the Test Session and Post Test were both conducted in-person at the Sony offices' gym in Lund.

The necessary equipment needed to conduct the tests are divided into a "Pre Test" and "Test Session & Post Test" categories:

Pre Test:

- Google Forms
- Doodle

Test Session & Post Test:

- A computer used for note-taking and data collection.
- An Android smartphone with the Hi-Fi prototype installed along with screen recording capabilities.
- An iPhone with the Hi-Fi prototype installed along with screen recording capabilities.
- Sony-specific NDA document for the test participant to sign.
- LTH-specific NDA document for the test participant to sign.
- Wooden stick (used to simulate a barbell)
- Dumbbell(s)
- Gym bench/Fitness bench
- The Camera-based Tracking System (CTS)
- Recording device (camera/smartphone) that records the main exercise area during the test session.
- A movie ticket for the test participant.

6.4.4 Test participants

The first test participant acted as a pilot test for the proposed usability test. After having completed the pilot test, parts of the test were iterated upon - in collaboration with the supervisor at Lunds Tekniska Högskola.

20 test participants was the number agreed upon by the supervisors. This meant that the participants would be evenly divided between the four *test scenarios*, meaning each scenario would be conducted five times (see Figure 6.19).

User group

As with Advagym and the Lo-Fi usability test - the system and solution to be tested had no defined user group.

The recruitment of test participants was aimed to be broad and diverse so as to not bias the usability test results with test participants who were experienced with applications that are used in gym environments. The only restrictions imposed in the recruitment process was that the test participants would need to be able to physically perform the following gym exercises (without pain or discomfort):

- Squats (with a wooden stick).
- Bicep Curl (with a dumbbell or wooden stick).
- Bench Press (with a wooden stick).

The test participant also needed to be able to see/interpret information and user interfaces displayed on a 6-inch smartphone screen.

Recruitment

Recruitment of the test participants was conducted through the use of a sign-up questionnaire (hosted by Google Forms), that was spread through different channels (student Facebook pages, a Crossfit gym in Lund along with personal contacts of Advagym employees and the project's supervisors).

An advertisement poster was also put up around IKDC (Ingvar Kamprad Design Centrum, located in Lund) and the rest of the Lunds Tekniska Högskola (see Appendix D).

An incentive to take part in the usability tests came in the form of a guaranteed movie ticket upon completion of a test session.

Attended participants

As mentioned above, 20 test participants were recruited and carried out the Hi-Fi usability test. The data and insights gained from the recruitment questionnaire (found in Appendix A) was used to see if there was a correlation found between the test participants gender, age, exercise activity, gym experience, etc. - and the usability of the prototypes.

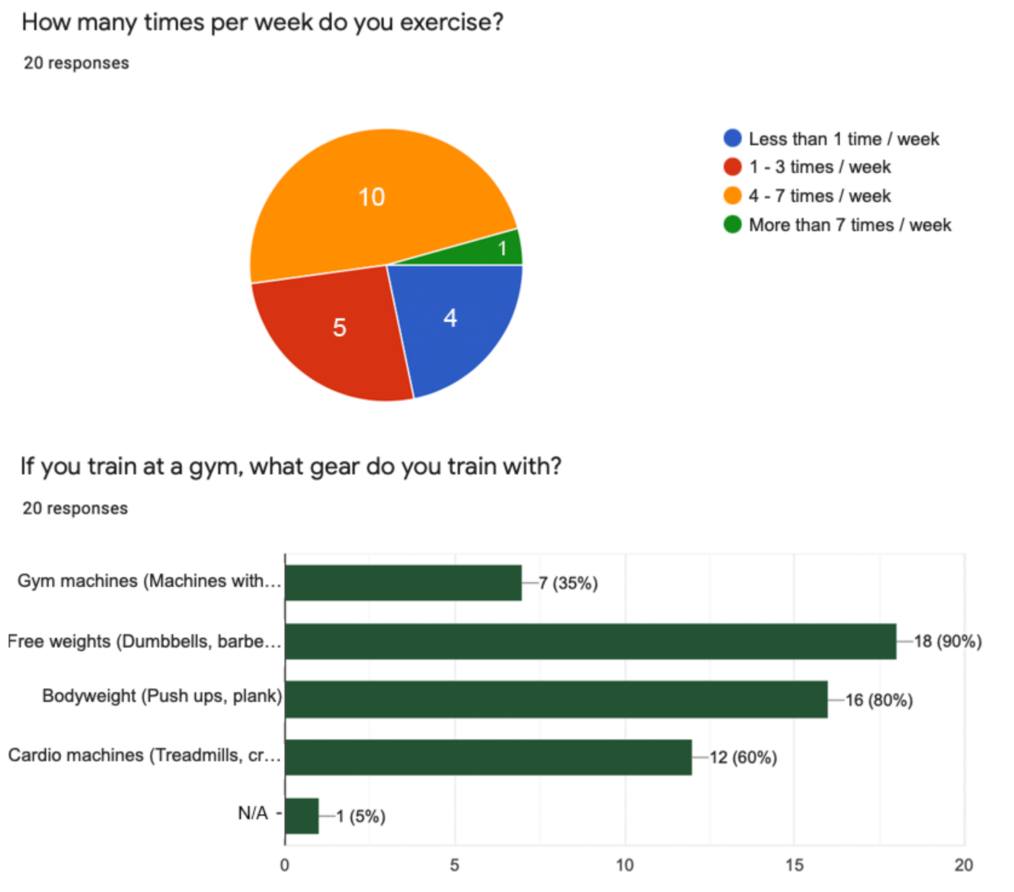


Figure 6.20: Distribution of exercise activity during a week, along with the distribution of equipment used/exercises performed when attending a gym.

The gender distribution of the participants was 11 males and 9 females. The average age of the participants was 32.6, within the span of 19 to 60 years old. The total weekly exercise activity of the test participants, along with what equipment/exercises they used when going to gym are found in Figure 6.20.

6.4.5 Procedure

Table 6.1: The required material, time estimates and descriptions for each part and step of the Hi-Fi usability test.

Part	Step	Description	Material	Time
Pre Test	**Sign Up / Online Questionnaire**	To register for the test, participants will fill out an online questionnaire.	Online Questionnaire	N/A
Pre Test	**Briefing**	Receive participant. Participant signs NDAs. Brief participant of the structure and background of the test	- Testing script - Non Disclosure Agreements	5-10 min
Test Session	**Task 1**	Carry out 2 sets of Squats, 5 repetitions each - with the help of the application and the automatic tracking feature.	- Testing script - Recording software - Smartphone - Barbell (with weights)	2-4 min
Test Session	Subtask 1.1	Connect to the CTS.	See Task 1	See Task 1
Test Session	Subtask 1.2	Understand that 2 sets, with 5 repetitions each, have been recorded correctly by the application.	See Task 1	See Task 1
Test Session	**Task 2**	Go over to a table outside of the tracked area - to get some water - and then to return to the exercise area.	- Testing script - Recording software	2-3 min
Test Session	Subtask 2.1	Understand that the tracking has been lost.	See Task 2	See Task 2
Test Session	**Task 3**	Carry out 1 set, 10 repetitions of Bicep Curls - with the help of the application and the automatic tracking feature.	- Testing script - Recording software - Smartphone - 2x Dumbbells	2-3 min
Test Session	Subtask 3.1	Reconnect to the CTS.	See Task 3	See Task 3
Test Session	Subtask 3.2	Understand that 1 set with 10 repetitions has been recorded correctly by the application.	See Task 3	See Task 3
Test Session	**Task 4**	Carry out 2 sets, 5 repetitions of Bench Press - with the help of the application and the automatic tracking feature.	- Testing script - Recording software - Smartphone - Barbell (with weights) - Gym bench	2-5 min
Test Session	Subtask 4.1	Reconnect to the CTS (if applicable).	See Task 4	See Task 4
Test Session	Subtask 4.2	Understand that 2 sets, with 5 repetitions each, have been recorded correctly by the application.	See Task 4	See Task 4
Test Session	**Task 5**	Carry out a couple of Bicep Curls repetitions - but do it without the application automatically tracking the repetitions/exercise.	- Testing script - Recording software - Smartphone	1 min
Test Session	**Task 6**	Carry out a couple of additional Bicep Curls repetitions - with the help of the application and the automatic tracking feature.	- Testing script - Recording software - Smartphone	1-2 min
Test Session	**Task 7**	Finish/end the workout.	- Testing script - Recording software - Smartphone	1 min
Post Test	**SUS questionnaire**	The questionnaire will let the participant score the usability of the user interface/system.	SUS questionnaire	2-5 min
Post Test	**Semi-structured interview**	A couple of questions will be asked where the participant is able to summarize their experience and provide some additional quantitative/qualitative data.	- Testing script - Recording software	2-5 min
			Total time:	~45 min

As previously mentioned - each task (Task 1, Task 2, etc.) was clearly communicated to the test participant. The subtasks (Subtask 1.1, Subtask 2.1, etc.) are to be considered a part of their respective parent task (Task 1, Task 2 etc.) - and were *not* communicated to the test participant. They were instead tracked and logged by the test moderator during the test session.

The exact procedure, materials required, along with the time estimates for the different parts and the total estimated duration of the test are detailed in Table 6.1. A testing script was also used in order for the test moderator to accurately reproduce the test for each participant. The Hi-Fi usability test script can be found under Appendix E.

6.4.6 Results

Test metrics

Looking at the test metrics found in Table 6.2 - one can observe the following distribution in terms of overall task completion/errors for each of the prototypes:

- **Hi-Fi prototype 1**:
  - *Success*: 90%
  - *Non-critical errors*: 6%
  - *Critical errors*: 4%
- **Hi-Fi prototype 2**:
  - *Success*: 88%
  - *Non-critical errors*: 10%
  - *Critical errors*: 2%
- **Hi-Fi prototype 3**:
  - *Success*: 92%
  - *Non-critical errors*: 4%
  - *Critical errors*: 4%
- **Hi-Fi prototype 4**:
  - *Success*: 88%
  - *Non-critical errors*: 10%
  - *Critical errors*: 2%

Table 6.2: This results and metrics for each task (and subtasks) concerning the Hi-Fi prototype test scenarios (i.e. Hi-Fi prototype 1, 2, 3 and 4).

Hi-Fi Prototype 1: Waypoint (no auditory feedback)					
Task	**Subtask**	**Success**	**Non-Critical Errors**	**Critical Errors**	**Time on Task (mean, seconds)**
Task 1	*Subtask 1.1*	3	2	0	184.6
	Subtask 1.2	4	1	0	
Task 2	*Subtask 2.1*	5	0	0	18.4
Task 3	*Subtask 3.1*	5	0	0	83
	Subtask 3.2	5	0	0	
Task 4	*Subtask 4.1*	4	0	1	98.6
	Subtask 4.2	4	0	1	
Task 5	N/A	5	0	0	21
Task 6	N/A	5	0	0	28
Task 7	N/A	5	0	0	11
Hi-Fi Prototype 2: Waypoint (with auditory feedback)					
Task	**Subtask**	**Success**	**Non-Critical Errors**	**Critical Errors**	**Time on Task (mean, seconds)**
Task 1	*Subtask 1.1*	1	4	0	197.6
	Subtask 1.2	4	0	1	
Task 2	*Subtask 2.1*	5	0	0	16.8
Task 3	*Subtask 3.1*	5	0	0	66.2
	Subtask 3.2	5	0	0	
Task 4	*Subtask 4.1*	5	0	0	147.2
	Subtask 4.2	5	0	0	
Task 5	N/A	5	0	0	34.4
Task 6	N/A	5	0	0	47
Task 7	N/A	4	1	0	9.8

Hi-Fi Prototype 3: Puck (no auditory and haptic feedback)					
Task	**Subtask**	**Success**	**Non-Critical Errors**	**Critical Errors**	**Time on Task (mean, seconds)**
Task 1	*Subtask 1.1*	3	1	1	157.8
	Subtask 1.2	3	1	1	
Task 2	*Subtask 2.1*	5	0	0	16.8
Task 3	*Subtask 3.1*	5	0	0	99.2
	Subtask 3.2	5	0	0	
Task 4	*Subtask 4.1*	5	0	0	132.4
	Subtask 4.2	5	0	0	
Task 5	N/A	5	0	0	42.4
Task 6	N/A	5	0	0	35.6
Task 7	N/A	5	0	0	15.4
Hi-Fi Prototype 4: Puck (with auditory and haptic feedback)					
Task	**Subtask**	**Success**	**Non-Critical Errors**	**Critical Errors**	**Time on Task (mean, seconds)**
Task 1	*Subtask 1.1*	4	1	0	153.8
	Subtask 1.2	4	1	0	
Task 2	*Subtask 2.1*	3	2	0	16
Task 3	*Subtask 3.1*	5	0	0	75.8
	Subtask 3.2	5	0	0	
Task 4	*Subtask 4.1*	5	0	0	117.6
	Subtask 4.2	5	0	0	
Task 5	N/A	5	0	0	21.8
Task 6	N/A	5	0	0	48.2
Task 7	N/A	3	1	1	19.2

SUS scores

Table 6.3: The resulting mean SUS score (out of 100) from the SUS questionnaires for each of the four Hi-Fi prototypes.

Prototype	SUS Score (mean):
Hi-Fi Prototype 1	**72.5**
Hi-Fi Prototype 2	**72**
Hi-Fi Prototype 3	**83**
Hi-Fi Prototype 4	**76**

20 different SUS-scores were compiled from the 4 prototypes tested and the 20 test participants. By comparing the mean SUS-scores (see Table 6.3) to the corresponding range found in the CGS table (see Table 3.1), a grade to describe the usability of the prototype was arrived at [50]:

- **Hi-Fi prototype 1** scored a 72.5, earning it a ***C+*** usability grade.
- **Hi-Fi prototype 2** scored a 72, earning it a ***C+*** usability grade.
- **Hi-Fi prototype 3** scored a 83, earning it a ***A*** usability grade.
- **Hi-Fi prototype 4** scored a 76, earning it a ***B*** usability grade.

Prototype 3 scored above 80, which indicates an *above average* user experience. The other prototypes (Prototypes 1, 2 and 4) all scored below 80, but above 71.1, which indicates an *average* user experience [53].

Semi-structured interview

During the Post Test semi-structured interview, 9 out of 20 test participants said that they would like to use a system such as the one proposed during the test. 9 out of 20 test participants said that they would like to use a system such as the one proposed during the test - *if* some changes were made/features were added, or if they had had more interest in going to gym/track their exercises in general. 2 out of 20 test participants said that they would *not* like to use a system such as the one proposed during the test.

That equates to an overall 90% of test participants being interested in using an automatic free weight exercise tracking solution.

A lot of feedback in regards to the prototypes' features/functionality and user interface was provided by the test participants. Examples of such feedback included:

- It was not clear that you were able to step out of the "waypoint" when performing exercises.
- It was not clear *where* and/or what the "waypoint" *was supposed to be*, in the gym.
- It was not clear if you had to bring your smartphone with you in order to be tracked by the system.

- It was not clear where the orange "puck" was, in the gym.
- The solution needed to be even more "automatic" (i.e. *less* interaction would be needed by its users).
- The repetition tracking was at times not accurate - and at other times did not work at all.
- The ability for the application to "guide" the user in the sense of offering programs with different exercises that the application would then "fill in" as the user was performing the different sets and repetitions.

Chapter 7

Discussion

This thesis has evaluated several modes of interaction and user interfaces interacting with a complex camera-based computer vision- and IoT-system.

The following sections in this chapter will discuss the working process and results thereof. The sections will also cover the research questions and to what extent the thesis answers them, the fulfillment of the project's sub-goals, the limitations and error sources present during the working process and what future work may be explored.

7.1 Lo-Fi usability test observations

The results of the Lo-Fi usability test were used to drive two important considerations: the first being the *narrowing* of the scope of the project, which had practical implications on the rest of the project. The second consideration had to do with the validation and evaluation of the *concepts* and *connection methods* derived from the ideation phase of the project:

1. Would the test participants of the Lo-Fi usability test understand *how to connect* to the CTS?
2. Would they understand *how to exercise* (i.e. interacting with the CTS), with the help of the prototype?
3. Would the test participants *understand* the CTS system and the automatic free weight exercise tracking concept as whole?
    - Do they even *need* to understand the system and concept - or is that irrelevant in regards to the usage thereof?

The results of the Lo-Fi usability test indicated that question 1 and 2 could be answered with a "yes", due to the above-average SUS scores in regards to the prototypes' usability, along with the low error rates observed (see Section 5.3.6).

The third question and its sub-question are harder to evaluate. The discussions and subjective data gained from the semi-structured interview conducted in the Post Test indicated that the test participants had a good understanding of what the system aimed to accomplish and what it could do for a potential user. This, due to the abundance of feature and functionality suggestions and feedback that every test participant provided.

Difficulty arose when it came to deciding which of the three prototypes (Lo-Fi prototype 1, 2 and 3) was the most compelling in terms of usability and user preference. All prototypes had very similar SUS scores and error rates, which inferred that *all* three prototypes were viable choices going forward. Therefore, the answers given by the test participants in the Post Test semi-structured interview became more important than previously anticipated.

The somewhat informal nature of the semi-structured interview provided a good sense of the test participants' understanding of the prototype and the CTS system and automatic free weight exercise tracking solution. However, the interview style also resulted in imprecise and ambiguous subjective data collection in regards to what prototype was most preferred. A subsequent ranking of the prototypes (see Figure 5.4) was therefore introduced, which in itself was not the most accurate representation of the collective test participants preference.

7.2 Hi-Fi usability test observations

The Hi-Fi usability test functioned as the last "activity" of the project: it was the final evaluation of all of the iterations made, the connection methods developed and implemented, and the underlying concepts proposed.

The results thereof, were somewhat mixed. Whilst the overall task completions of the four prototypes (Hi-Fi prototype 1, 2, 3 and 4) were marked 89.5% *success*, 7.5% *non-critical error* and 3% *critical error* - the mean SUS scores were not impressive in comparison to the scores received by the Lo-Fi prototypes (see Section 6.4.6 and Section 5.3.6). How did this come about?

There are many potential answers to that question. For one, the CTS's object (human) tracking and exercise detection does not have 100% accuracy. When repetitions were not detected by the CTS, the repetition counter displayed on the prototype or communicated to the user by way of audio - was not incremented. *Task 4* of the usability test instructed the user to go to the far corner of the gym and do sets of Bench Press exercises. This activity was specifically chosen since prior knowledge of the CTS in the gym had shown that, that far corner combined with the Bench Press exercise led to the system losing track of the user and therefore not registering their repetitions. The non-registered repetitions and loss of tracking frustrated several test participants and might have led them to believe that the prototype and system was difficult to use and complex, which was not the intended purpose of the task.

The addition of more feedback in the form of auditory and haptic feedback in the prototypes (Hi-Fi prototypes 2 and 4) did *not* correlate into higher mean SUS scores or higher successful task completion in comparison to their counterparts without such feedback (Hi-Fi prototypes 1 and 3, see Section 6.4.6). This was not anticipated, and trying to explain why this is - is hard to quantify. One possibility could be that the modes of feedback did not have high enough *affordance*, or was not sufficiently explained/put into context for the user. On the other hand, several test participants that used the prototypes without the additional feedback (Hi-Fi prototypes 1 and 3) explicitly asked for the addition of such forms of feedback during their Post Test semi-structured interviews. This indicated that such additions would have positive effects on the overall usability of the prototype.

It was also clear that the small outlined circle on the floor in the gym, representing the "waypoint", needed to have more *discoverability* and perhaps be afforded more *mapping* in relation to free weights. This could be done with additional signage, or making sure that the illustration on the Exercise screen was mapped to the circle on the floor (i.e. looking more or less identical). Some of the test participants did not find the waypoint, and some participants also chose to stand within the circle on the floor during their exercises. The *conceptual model* of the prototype had therefore not been sufficiently developed, so that the user would know that they were free to move around the gym area when performing exercises. Several users thought that it would be a good idea to have a more thorough and comprehensive "tutorial" (onboarding) of how the system functioned before using the system for the first time.

The prototypes with the highest mean SUS scores of the Hi-Fi usability test, were the prototypes that used the "puck" connection method (Hi-Fi prototype 3 and 4). People in general are already quite experienced with "tapping" on things in different contexts of their lives: such as when tapping one's credit card to a register when paying at a store, or when tapping one's transit card when using public transport. This perhaps made this form of interaction feel more *intuitive* and natural to the user. Building upon already-established interactions could prove useful when users need to interact with a totally new piece of technology/system.

The Hi-Fi prototypes procured and the concepts proposed are considered successful in terms of overall user satisfaction, since 90% of test participants said that they would like to use such an automatic free weight exercise tracking system.

7.3 Research questions

Several research questions were posed at the start of this thesis work. Through the use of the data and results collected, they can now be discussed and/or answered:

What are the best practices in regards to designing user-facing interfaces for 3D object-tracking camera-based computer vision systems?

Evaluation of the Hi-Fi prototypes has shown that the two main aspects to consider when designing these user interfaces are:

1. Clearly explain to the user how you interact with the *physical* aspects of the system.
2. There has to be an abundance of *feedback* available to the user.

This thesis proposes that in order to (intentionally) connect and interact with such a system, there should be a *physical* element to the initial interaction. This interaction needs to be explained and shown in detail to the user, since it functions as the gateway to the entire system/solution.

Since the system is "invisible" to the user, the user has no insight into the system's current perception or status. Therefore, the system needs to effectively communicate with the user; and the best way of doing this is through an abundance of *feedback* whenever the user interacts with the system. This includes auditory feedback as well as other forms of feedback that can successfully draw a user's *attention*.

How do you evaluate and analyze the robustness and efficacy of such a user interaction, with the system?

The designed user interactions, concepts and solutions for systems of the kind discussed in this thesis work need to go through extensive testing. Simulation of such systems during testing is not preferable, since there might be several technical hurdles/constraints involved with the final, real-world implementation. Some of these constraints might impact the proposed concepts or solutions, making some of them hard or not possible to contend with. Early testing is always good practice in large projects - but the *later* testing is possibly more important when it comes to these kinds of systems due to potentially novel concepts considered during development.

Usability tests are a good testing methodology to use when evaluating user interactions and concepts. It helps you identify the fundamental problems with the concept, design or interaction you are evaluating. Since the type of system we're examining is heavily user-focused, usability testing helps with learning about said user's behavior and preferences whilst interacting with the system. The feedback gained from the usability test participants can then be used to iterate and improve the system and solution in a - for the end user - meaningful way.

How do you offer and visualize onboarding, feedback, user flows and different interaction modes of such a system, to a user?

The Hi-Fi usability test results conclusively showed that there were still issues with the onboarding and feedback of the Hi-Fi prototypes. As such, it has been shown that it is of utmost importance to put a *lot* of time and effort into the scope, design and development of these visualizations.

The onboarding of these systems needs to be extensive, and should be aided by the use of *animations* or *videos* that better *map* to the physical parts of the system. Feedback needs to be designed to grab the user's *attention*, since interacting with such systems might imply that the user is not constantly looking down at a display of some kind.

Which of the proposed approaches/solutions/prototypes is the most compelling, from a real-world business perspective (specifically for integration into Advagym's current systems)?

The Hi-Fi prototypes had very similar performance when it came to successful task completion. However, the highest mean SUS scores were awarded to the prototypes using the "puck" connection method (Hi-Fi prototype 3 and 4).

That, together with the fact that Advagym already uses the puck as a means of interacting with the IoT devices currently in their system - meant that the "puck" connection method would be of most interest for them, going forward.

7.4 Project sub-goals

Since the thesis work and project also should result in something of direct use to Advagym and Sony's R&D Center Lund Laboratory - a couple of additional specific sub-goals were put forward. To what extent these goals were met, is detailed below:

Implement a prototype application that runs on both Android and iOS devices.

This goal was successfully met, through the use of the React Native framework that allowed for a single code base that ran on both the iOS and Android operating system/devices.

The prototype application shall be able to interact and communicate with Advagym's IoT devices and APIs (no simulation).

None of the features or functionalities of the application was simulated. The application is also able to interact and communicate with Advagym's puck and APIs. This meant that this goal was met.

The prototype application shall also be able to interact with Sony's R&D Center Lund Laboratory CTS (no simulation).

As with the previous goal - the application is able to interact with the CTS, with no simulation. This meant that this goal was also met.

Internal testing and debugging of Advagym and Sony's R&D Center Lund Laboratory's systems shall be possible.

This goal was also fully met. The application is currently available for debugging and evaluation purposes, for Advagym and Sony's R&D Center Lund Laboratory team members.

The application is also available to test for any employee working at the Sony offices in Lund. Posters are hung in the gym, detailing how to install and use the application.

Data from the testing and debugging shall be exportable and easily parsed.

Through the use of Slack Webhooks, logging and feedback data related to the CTS is exported and able to be accessed and parsed by R&D team members - meaning that this goal was met.

7.5 Limitations and error sources

Design and work process

Every step of the design and working process applied during the project was continuously communicated and discussed with the Advagym supervisor (and less frequently - also with

Lunds Tekniska Högskola's supervisor). However, the thesis was just conducted by a single individual - meaning that activities such as ideation (brainstorming/bodystorming) did not receive continuous input or differing perspectives as it would've with a group/pair project. That could mean that a lot of problems with the concept and solutions proposed might have been avoided or approached/designed differently.

Another aspect of only having a single individual working on the project, meant that the scope of the project needed to be narrower than if there had been more individuals working together. When it came to development and implementation - that also meant that the project had a heavy reliance on the technical knowledge of that single individual. This in turn could have had effects on the design decisions considered, in terms of what functionality and features the prototypes would implement - since there were a lot of technologies that the individual had no prior experience with.

Usability testing

The higher mean SUS scores and test results of the Lo-Fi usability testing could have been impacted by the fact that none of the test sessions pertaining to it were done in-person in a gym due to the COVID-19 pandemic. Not carrying out the Lo-Fi usability test in-person could also be one of the reasons behind the discoverability problems discovered when test participants were trying to find the "waypoint" and "puck" during the later Hi-Fi usability test.

During the usability test sessions of the Hi-Fi prototype - some participants were more or less distracted. Since the test sessions were conducted in the Sony office gym - that meant that other employees could walk in at any moment in order to work out or test gym equipment. Some test participants had no other persons present in the gym whilst conducting the test sessions, and some test participants had several (up to four).

The Hi-Fi usability test session itself had a major design flaw. *Task 5* instructed the user to perform Bicep Curls exercises, but without being connected and tracked by the system. The intention was to evaluate the user's ability to disconnect from the tracking system. But since the preceding task (Task 4) instructed the user to perform Bench Presses in the far corner of the gym - this *almost* always meant that the user would disconnect from the system due to the system losing track of the user. What this in turn meant, was that when the user was asked to complete Task 5 - they were already disconnected from the system. This meant that only 2 out of 20 test participants interacted with the connection status button (seen in the top-right corner of the last screen in Figure 6.11). This in turn might have had a clear impact on the user's ability to understand when they were connected to the system, and when they were not - negatively impacting the overall understandability of the system.

Another flaw of the usability testing had to do with the task completion definitions with their associated fail/pass criteria. The definition of *success* and *non-critical error* were at times ambiguous. What constitutes "completed with minimal assistance"? It would have to be up to interpretation. However, since only a single individual - acting as the test moderator - recorded the task completion, there was at least consistency in that regard.

Implementation and technologies used

The prototype application is fully reliant on what the CTS detects and communicates through its broker by way of MQTT messages. If the user performs a repetition that isn't detected by the CTS, the application shows no feedback to indicate this - since there is no way of knowing what the user is doing. This could have led to some of the negatively perceived aspects of the application, even though it wasn't an app-related issue.

The state handling implemented in the application made the messages from the CTS available to all parts of the system. On Android devices, the performance of the state handler (React Redux) was not as fast on iOS devices - meaning that feedback received from completing a repetition was slower on Android than on iOS. Some users showed signs of acknowledging this fact by slowing their repetition pace whilst looking at the smartphone display. This might've added bias to the Hi-Fi usability test results since test participants had the choice of using either an Android or iOS device during the test session.

7.6 Future work

There are plenty of additional approaches, ideas and avenues to explore when it comes to improving and exploring the underpinnings of a solution for interaction with a complex camera-based computer vision- and IoT-system. Some examples of these are:

- Animations and videos could be used to help the user understand aspects of the system and solution better than static images and text.
- There might be ethics and privacy concerns related to camera-based tracking systems that were not explored in this project.
- Several test participants voiced a disdain for having to hold and interact with their smartphone device during workouts; perhaps a solution involving some form of wearable device could be explored.
- Currently, only repetitions and sets of exercises, are tracked. Another big part of the logging gym-goers spend time on is that of weights. Perhaps the automatic free weight exercise tracking solution could support automatic weight detection as well.
- Additional user guidance (i.e. some form of "virtual PT") could be implemented in the form of a voice talking to and guiding the user through the entirety of a workout with multiple different exercises - instead of the user having to manually go to each exercise to log the sets and repetitions.
- Making the entire automatic free weight exercise tracking solution *fully* automatic. This could mean that the user only needs to connect to the CTS once at the start of a workout, and then all exercises would be logged by the application automatically. No further user input needed.
- The CTS could be used to help the user with correct posture when it comes to performing exercises through its pose estimation.

Chapter 8

Conclusion

As stated in Section 1.1 - the gym-going experience is often a complicated affair with a lot of effort needed if one wants to perform at one's best. A large part of the effort expended goes into monitoring and logging the results of performed workouts and its exercises. The user interface, automatic free weight exercise tracking solution and connection methods proposed in this thesis were aimed to make this experience *easier* and require less *effort*. If a solution such as the one proposed can encourage and entice people with or without prior "gym experience", into attending a gym - then it will have a positive impact on helping people exercise and stay healthy.

The fields of computer vision and machine learning are firmly on track to further revolutionize fields that aren't specifically computer science-related. The fact that an automatic free weight exercise tracking solution is currently within reach, is a fascinating proposition for other related applications. There is much research and experimentation ahead, when it comes to designing user interfaces and interactions that work well with solutions and systems of this kind.

The test participants that used and evaluated the prototypes created in the project have shown a high level of understanding of the automatic free weight exercise tracking system - and 90% of them would like to use a system such as the one proposed if it existed. Therefore, the prototypes created during this project have shown that such user interfaces and solutions are not only *possible*, but already function *well*.

References

[1] K. L. Piercy, R. P. Troiano, R. M. Ballard, S. A. Carlson, J. E. Fulton, D. A. Galuska, S. M. George, and R. D. Olson. The physical activity guidelines for americans. *JAMA*, 320(19), 11 2018.

[2] Statista. Market size of the global health club industry from 2009 to 2019, May 2020. https://www.statista.com/statistics/275035/global-market-size-of-the-health-club-industry/ [Accessed: 2022-04-19].

[3] Grand View Research. Fitness app market size, share trends analysis report by type (exercise weight loss, diet nutrition, activity tracking), by platform (android, ios), by device, by region, and segment forecasts, January 2022. https://www.grandviewresearch.com/industry-analysis/fitness-app-market [Accessed: 2022-04-19].

[4] G. Jayavardhana, B. Rajkumar, M. Slaven, and Marimuthu P. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[5] J. Håkansson. Interaction with iot data to help users train smarter. 2019.

[6] J. Rudberg and D. Motylinska. Designing and evaluating a free weight training application. 2021.

[7] United Nations. Agenda 2030, 2015. https://sdgs.un.org/2030agenda [Accessed: 2022-04-20].

[8] E. Dave. How the next evolution of the internet is changing everything. *The Internet of Things*, 2011.

[9] Statista. Number of internet of things (iot) connected devices worldwide from 2019 to 2030, December 2020. https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ [Accessed: 2022-04-28].

[10] Google Cloud. Technical overview of internet of things, 2016. `https://cloud.google.com/architecture/iot-overview` [Accessed: 2022-04-28].

[11] MQTT. `https://mqtt.org` [Accessed: 2022-04-28].

[12] U. Hunkeler, H. L. Truong, and A. Stanford-Clark. Mqtt-s — a publish/subscribe protocol for wireless sensor networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, pages 791–798. IEEE, 2008.

[13] T. Yokotani and Y. Sasaki. Comparison with http and mqtt on required network resources for iot. In *2016 international conference on control, electronics, renewable energy and communications (ICCEREC)*. IEEE, 2016.

[14] I. El Naqa and M. J. Murphy. What is machine learning? In *machine learning in radiation oncology*, pages 3–11. Springer, 2015.

[15] G. Bonaccorso. *Machine learning algorithms*. Packt Publishing Ltd, 2017.

[16] P. P. Shinde and S. Shah. A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, pages 1–6. IEEE, 2018.

[17] S Prince. *Computer vision*. Cambridge University Press, Cambridge, England, June 2012.

[18] OpenCV. `https://opencv.org/about/` [Accessed: 2022-04-28].

[19] J. Scharcanski and M. E. Celebi. *Computer vision techniques for the diagnosis of skin cancer*. Springer, 2013.

[20] C. Zhu. Multi-camera people detection and tracking, 2019.

[21] R. Mohedano, C. R. Del-Blanco, F. Jaureguizar, L. Salgado, and N. García. Robust 3d people tracking and positioning system in a semi-overlapped multi-camera environment. In *2008 15th IEEE International Conference on Image Processing*, pages 2656–2659. IEEE, 2008.

[22] L. Chen, H. Ai, R. Chen, Z. Zhuang, and S. Liu. Cross-view tracking for multi-human 3d pose estimation at over 100 fps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3279–3288, 2020.

[23] React Native. `https://reactnative.dev/` [Accessed: 2022-04-28].

[24] Flutter. `https://flutter.dev` [Accessed: 2022-04-28].

[25] W. Scacchi. Is open source software development faster, better, and cheaper than software engineering. In *2nd Workshop on Open Source Software Engineering, Orlando, Florida*, volume 3, page 2, 2002.

[26] A. Polacco and K. Backes. The amazon go concept: Implications, applications, and sustainability. *Journal of Business and Management*, 24(1):79–92, 2018.

[27] K. Wankhede, B. Wukkadada, and V. Nadar. Just walk-out technology and its challenges: A case of amazon go. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 254–257. IEEE, 2018.

[28] R. Khurana, K. Ahuja, Z. Yu, J. Mankoff, C. Harrison, and M. Goel. "gymcam: Detecting, recognizing and tracking simultaneous exercises in unconstrained scenes". *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(4), dec 2018.

[29] Strong. https://www.strong.app [Accessed: 2022-05-02].

[30] Jefit. https://www.jefit.com [Accessed: 2022-05-02].

[31] FitNotes. http://www.fitnotesapp.com [Accessed: 2022-05-02].

[32] P. Krafft, J. Zheng, E. Shmueli, N. Della Penna, J. Tenenbaum, and S. Pentland. Popularity and performance: A large-scale study. *arXiv preprint arXiv:1406.7729*, 2014.

[33] D. Yoganathan and S. Kajanan. Persuasive technology for smartphone fitness apps. In *PACIS*, page 185. Citeseer, 2013.

[34] D. Norman. *The design of everyday things: revised and expanded edition*. Basic Books, London, England, 2 edition, November 2013.

[35] J. D. Gould and C. Lewis. Designing for usability: key principles and what designers think. *Communications of the ACM*, 28(3):300–311, 1985.

[36] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta. Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*, 2017.

[37] D Norman. The four fundamental principles of human-centred design and application. 2019. https://jnd.org/the-four-fundamental-principles-ofhuman-centered-design/ [Accessed: 2022-05-28].

[38] C. Wasson. Ethnography in the field of design. *Human organization*, 59(4):377–388, 2000.

[39] C. Magnusson, K. Rassmus-Gröhn, K. Tollmar, and E. Deaner. User study guidelines. https://www.certec.lth.se/fileadmin/certec/publikationer/HaptiMap_d12.pdf [Accessed: 2022-05-04].

[40] O. Serrat. The scamper technique. In *Knowledge Solutions*, pages 311–314. Springer, 2017.

[41] A. M. Davis. Software prototyping. In *Advances in computers*, volume 40, pages 39–63. Elsevier, 1995.

[42] J. Rudd, K. Stern, and S. Isensee. Low vs. high-fidelity prototyping debate. *interactions*, 3(1):76–85, 1996.

[43] AdobeXD. https://www.adobe.com/products/xd.html [Accessed: 2022-05-04].

[44] Figma. https://www.figma.com/about/ [Accessed: 2022-05-04].

[45] Sketch. https://www.sketch.com [Accessed: 2022-05-13].

[46] C. Kaner. A tutorial in exploratory testing. *Tutorial presented at QUEST2008.(Available online at: http://www. kaner. com/pdfs/QAIExploring. pdf, accessed: 26 Jan 2014)*, 2008.

[47] J. R. Lewis. Usability testing. *Handbook of human factors and ergonomics*, 12:e30, 2006.

[48] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008.

[49] J. Brooke. Sus: a "quick and dirty'usability. *Usability evaluation in industry*, 189(3), 1996.

[50] J. Sauro and J. R. Lewis. *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann, 2016.

[51] J. R. Lewis. The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction*, 34(7):577–590, 2018.

[52] Google Meet. `https://apps.google.com/meet/` [Accessed: 2022-05-17].

[53] J. R. Lewis and J. Sauro. Item benchmarks for the system usability scale. *Journal of Usability Studies*, 13(3), 2018.

[54] TypeScript. `https://www.typescriptlang.org` [Accessed: 2022-05-16].

[55] npm. `https://www.npmjs.com` [Accessed: 2022-05-16].

[56] React Native: Hot Reloading. `https://reactnative.dev/blog/2016/03/24/introducing-hot-reloading` [Accessed: 2022-05-16].

[57] React Redux. `https://react-redux.js.org/introduction/getting-started` [Accessed: 2022-05-16].

[58] Mosquitto. `https://mosquitto.org` [Accessed: 2022-05-16].

[59] iBeacon. `https://developer.apple.com/ibeacon/` [Accessed: 2022-05-16].

[60] react-native tts. `https://github.com/ak1394/react-native-tts` [Accessed: 2022-05-16].

[61] Axios. `https://axios-http.com/docs/intro` [Accessed: 2022-05-16].

[62] Slack Webhooks. `https://api.slack.com/messaging/webhooks` [Accessed: 2022-05-17].

[63] Slack. `https://slack.com` [Accessed: 2022-05-17].

[64] GitHub. `https://github.com/` [Accessed: 2022-05-17].

[65] GitHub Issues. `https://github.com/features/issues` [Accessed: 2022-05-17].

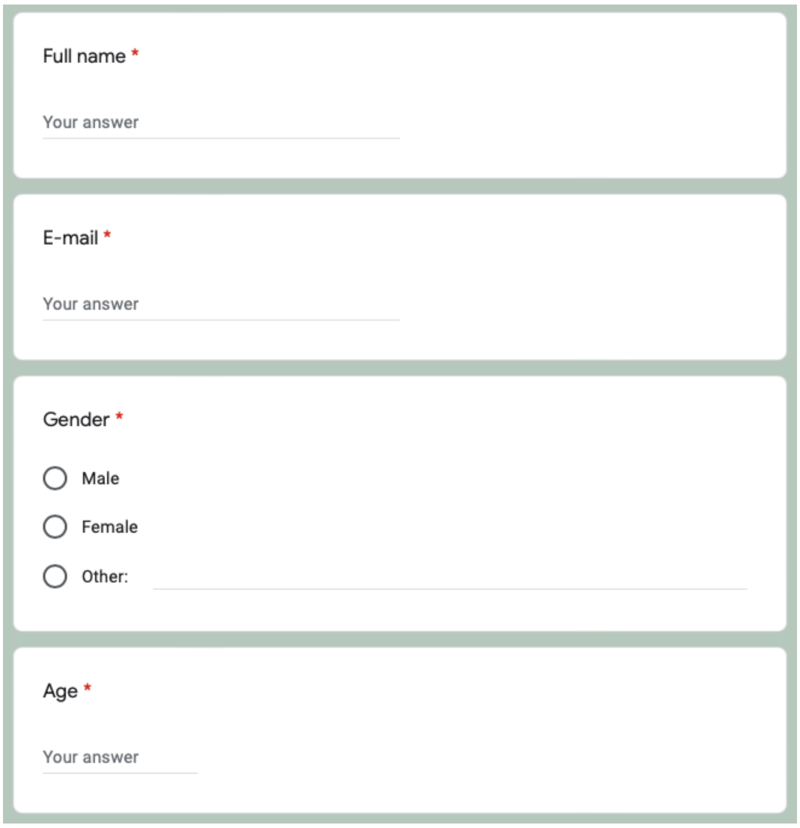
[66] Git. `https://git-scm.com/about` [Accessed: 2022-05-17].

Appendices

Appendix A

Lo-Fi and Hi-Fi usability test recruitment questionnaire

This was the recruitment questionnaire with its questions, sent out to test participants ahead of the usability test session for the Lo-Fi prototype:


Full name *

Your answer

E-mail *

Your answer

Gender *

- Male
- Female
- Other:

Age *

Your answer

Physical effort of everyday employment (How much do you need to work with your body and physically effort in your job / studies) *

	1	2	3	4	5	
Very little / none	○	○	○	○	○	Very much

How many times per week do you exercise? *

- ○ Less than 1 time / week
- ○ 1 - 3 times / week
- ○ 4 - 7 times / week
- ○ More than 7 times / week

How long have you kept up with your current training frequency? *

- ○ 1-3 months or less
- ○ 3-6 months
- ○ 5-12 months
- ○ 1-2 years
- ○ 2 years or more

What kind of exercise do you perform? Please specify: (ex: football, cycling, strength training, group training, etc.) *

Your answer

If you train at a gym, what gear do you train with? *

- [ ] Gym machines (Machines with added weights)
- [ ] Free weights (Dumbbells, barbells)
- [ ] Bodyweight (Push ups, plank)
- [ ] Cardio machines (Treadmills, crosstrainers)
- [ ] Other: ______

If you have some form of visual impairment, do you use any equipment to correct this, during training? For example lenses / glasses. *

- ○ Yes, I wear lenses / glasses
- ○ No, I don't wear lenses / glasses
- ○ I have no visual impairment

Do you have any movement reduction that prevents you from performing any certain movements? If so, please specify:

Your answer ______

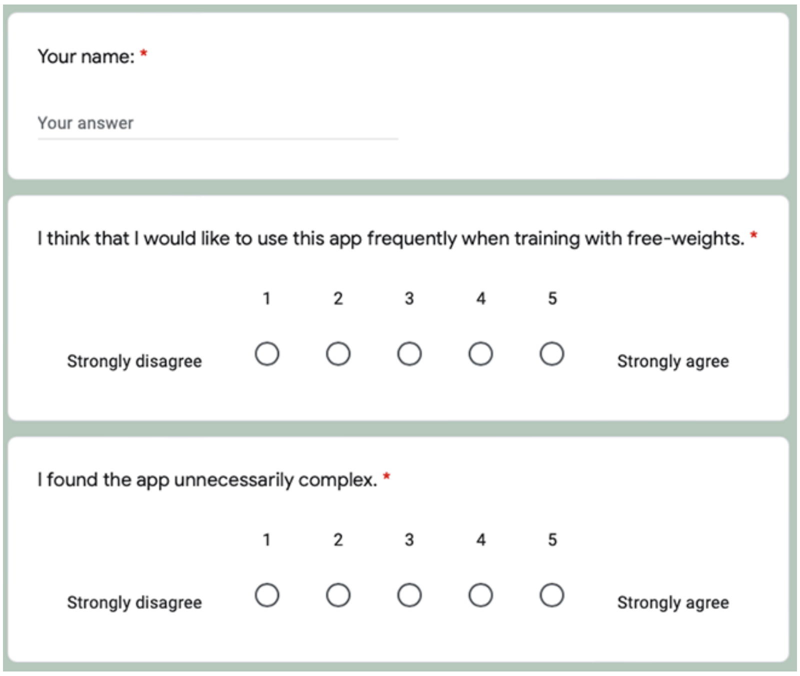
Which days and times of the week could you consider participating in the test? The test is on one occasion and takes about 30 minutes. Click on the link below to sign up.

https://doodle.com/meetme/qc/sgSpEGMvaM

Appendix B

Lo-Fi and Hi-Fi usability test SUS questionnaire

This was the SUS questionnaire presented to the test participants after having completed each of the three test scenarios of the Lo-Fi usability test:



Your name: *

Your answer

I think that I would like to use this app frequently when training with free-weights. *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I found the app unnecessarily complex. *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I thought the app was easy to train with. *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I think that I would need the support of a technical person to be able to train with this app. *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I found the various functions in this app were well integrated. *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I thought there was too much inconsistency in this app. *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I would imagine that most people would learn to use this app very quickly. *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I found the app very cumbersome to use. *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I felt very confident using the app. *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

I needed to learn a lot of things before I could start training with the app. *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

Appendix C

Lo-Fi usability test script

Task	Comment
Send invite-link to Teams-call & setup sketch prototype	Send out a secure link to a Microsoft Teams video call in order to conduct the test. PUCK: https://www.sketch.com/s/6f2707fe-e9a0-4d81-93f6-9cc90a2ee173/a/PG1JWR8/play https://forms.gle/JqfbaMeYmhw7Hush6 WAYPOINT: https://www.sketch.com/s/6f2707fe-e9a0-4d81-93f6-9cc90a2ee173/a/nRQvjd4/play https://forms.gle/SJWeorvD4bwTrfh98 GESTURE: https://www.sketch.com/s/6f2707fe-e9a0-4d81-93f6-9cc90a2ee173/a/kaKx73g/play https://forms.gle/ELE2Lafyd4GqLEmT9
Introduction	1). Background about Advagym. 2). Explain the structure of the test: A). 3 different "systems"/"prototypes" to test B). SUS - Questionnaire C). Short interview 3). Sign: NDA 4). Additional information: - It's the system we're testing, not you. - You're free to leave at any time, but we prefer if you complete the test. - You're free to ask any questions, but we might not be able to answer all of them until after the test. - Ask if recording the interactions made on the prototype along with audio of the call - not the participants webcam/etc. - is acceptable to the participant.
1. Prepare the Lo-Fi Sketch Prototypes. 2. Share your screen for the test particpant (Start recording the prototype screen and sound - if acceptable for participant) 3. Choose the next scenario and its prototype: A, B or C. 4. Depending on the prototype - ask the participant to place an object in front of them that will function as a "puck" - or tell them about the markings on the gym free weight area's floor. 5. Inform the test participant that they are able to control the prototype themselves - but that you might be controlling the prototype to simulate a system response/feedback.	

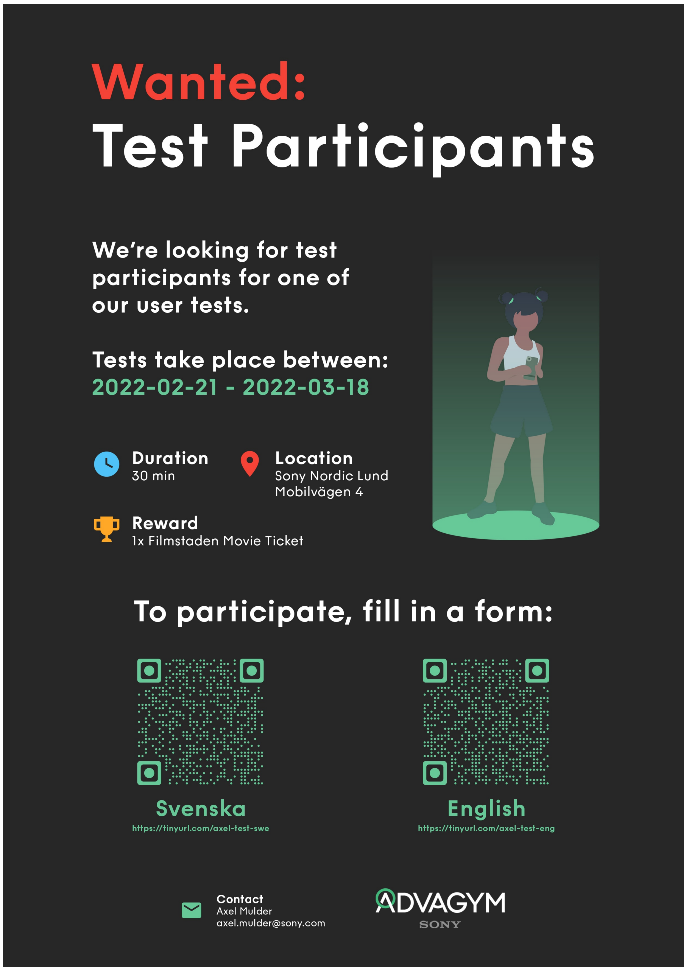
Test: A **Gesture**	Ask the participant to "think out-loud" of how they think they are supposed to connect to the tracking system. Ask the participant to start doing a deadlift exercise through the application. Walk the participant through simulating a tap on a "connect" button on their phone - and then doing a specific gesture. Ask the participant to begin an exercise. Ask the participant to "walk away to get some water". [The system has now lost contact with the participant.] Ask the participant to reconnect to the system. Ask the participant to disconnect from the system manually. Ask the participant to reconnect to the system. Ask the participant whether they are currently connected to the system or not. Ask **why/why not** they think they are currently connected to the system.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.
Test: B **Waypoint**	Ask the participant to "think out-loud" of how they think they are supposed to connect to the tracking system. Ask the participant to start doing a deadlift exercise through the application. Walk the participant through going to a specific point in the room, and ask them to simulate tapping the "connect"-button on their phone. Ask the participant to begin an exercise. Ask the participant to "walk away to get some water". [The system has now lost contact with the participant.] Ask the participant to reconnect to the system. Ask the participant to disconnect from the system manually. Ask the participant to reconnect to the system. Ask the participant whether they are currently connected to the system or not. Ask **why/why not** they think they are currently connected to the system.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.

Test: C **Puck**	Ask the participant to "think out-loud" of how they think they are supposed to connect to the tracking system. Ask the participant to start doing a deadlift exercise through the application. Walk the participant through tapping an object in the room with their phone in order to simulate tapping a puck with their phone. Ask the participant to begin an exercise. Ask the participant to "walk away to get some water". [The system has now lost contact with the participant.] Ask the participant to reconnect to the system. Ask the participant to disconnect from the system manually. Ask the participant to reconnect to the system. Ask the participant whether they are currently connected to the system or not. Ask **why/why not** they think they are currently connected to the system.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.
Choose next prototype/scenario: **A or B or C**	
Test: A **Gesture**	See above.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.
Test: B **Waypoint**	See above.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.
Test: C **Puck**	See above.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.
Choose next prototype/scenario: **A or B or C**	
Test: A **Gesture**	See above.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.
Test: B **Waypoint**	See above.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.

Test: C **Puck**	See above.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.
Choose next prototype/scenario: **A or B or C**	
Test: A **Gesture**	See above.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.
Test: B **Waypoint**	See above.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.
Test: C **Puck**	See above.
SUS Questionnaire	Ask the participant to fill in the SUS questionnaire regarding the just completed prototype.
Test session complete. Stop the recording. Inform the participant that recording has ceased.	
Short interview	Questions: 1). Have you used any gym training application before? If so: which ones? 2). Using the application and the tracking system - which of the 3 prototypes/scenarios (puck, waypoint, gesture) during the test session, would you prefer? 3). Having tested the application - would you like to use a system that tracks your exercises, like the one proposed? Why, why not? 4). Any other comments?
Finish	1). Thank the participant and allow for any other questions. 2). Say goodbye, close the video call.

Appendix D

Hi-Fi usability test recruitment poster



Appendix E

Hi-Fi usability test script

Task	Comment
Introduction & Briefing	
Briefing	1. Background information regarding Advagym. 2. Explain and ask the following (if needed): a. Can you define what sets/reps are? b. How the exercises Bicep Curls, Squats and Bench Press, work. Clarify that Bicep Curls are done one arm at a time. c. Ask if the test participant wants a wooden stick - or a real barbell whilst doing the exercises. d. Explain what *tracking* means: "The system understands and sees your movements." 3. Explain the structure of the test: a. You will perform specific tasks. b. After that - answer a short SUS questionnaire. c. Lastly, a short interview. 4. Additional information: a. It's the system we're testing, not you. b. You're free to leave at any time, but we prefer if you complete the test. c. You're free to ask any questions, but we might not be able to answer all of them until after the test. d. Ask if recording the interactions made on the prototype along with a video recording of the exercises - is acceptable to the participant. e. Would you like an iPhone or Android device? f. Ignore the colors and text that are present in the top system row on the device. 5. Give a little "background story" a. You want to perform a couple of exercises at the gym. b. You've seen that there's an application for your smartphone that automatically registers your reps and sets - on the Google Play Store/the AppStore. c. You download and install the application, create an account and login. d. Now it's time to start working out! e. Think out-loud when performing the tasks.
Test Session 1. Make sure you've turned on the video recording 2. Turn off the TV with the visualization on it. 3. Make sure that the device loaned out doesn't auto-lock. 4. Make sure the on-device recording is activated. 5. Bring up the Sheets-document with the usability test results. 6. Fetch a wooden stick and make sure that the fitness benches are in the correct positions.	
Task 1	Now I want you to stand here in front of the mirror - and imagine that you have 50kg to lift - and then I want you to do 2 sets of Squats, 5 repetitions in each - using the app and the automatic tracking feature. Tell me when you feel you are done.
Task 2	That was a couple of really tough sets! You have suddenly become very thirsty. I want you to go to the table over here and drink some water. Then you can go back to the training area. Tell me when you feel you are done.
Task 3	Now I want you to go to the mirror here - and do 1 set, 10 repetitions of Bicep Curls - using the app and the automatic tracking feature. Tell me when you're done.

Task 4	Now I want you to go to the bench over there in the far-right corner - and do 2 sets of Bench Press, 5 repetitions in each - using the app and the automatic tracking feature. Tell me when you feel you are done.
Task 5	Now you feel like you want to do a couple of single Bicep Curls over here by the mirror - but you do not want to be tracked by the system. Tell me when you feel you are done.
Task 6	Now that you've done a pair of Bicep Curls - now I want you to start the automatically tracking exercises again! Tell me when you feel you are done.
Task 7	Now I'd like for you end your workout for the day. Tell me when you feel you are done.
The test session is now over. 1. End the video recording and the screen recording. 2. Inform the test person that all forms of recording have been completed. 3. Make sure you have filled in everything that needs to be filled in, in the Sheets-document. 4. Start the audio recording of the short interview	
SUS questionnaire	Ask the test person to fill in the SUS questionnaire.
Short interview	Questions: 1. Have you used any gym application before? If so, which one(s)? 2. What are your thoughts regarding the connection and tracking of the system? 3. Did you feel that you understood when you were - and when you were not - connected to the system? Did you feel in control? 4. What are your thoughts on training with the system? 5. Do you think the tracking system behaved as you had expected? 6. Do you have any improvement suggestions or further wishes? 7. Any other comments?
End	1. Thank the test person and leave the floor open for other questions. 2. Give the test participant a movie ticket. 3. Say goodbye and follow the person up and out of the building.

Subtasks	
Subtask 1.1	Connect to the CTS.
Subtask 1.2	Understand that 2 sets, with 5 repetitions each, have been recorded correctly **or** incorrectly, by the application.
Subtask 2.1	Understand that the tracking has been lost.
Subtask 3.1	Reconnect to the CTS.
Subtask 3.2	Understand that 1 set with 10 repetitions has been recorded correctly **or** incorrectly, by the application.
Subtask 4.1	Reconnect to the CTS (if applicable).
Subtask 4.2	Understand that 2 sets, with 5 repetitions each, have been recorded correctly **or** incorrectly, by the application.