ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

CORSO DI LAUREA MAGISTRALE IN
INGEGNERIA INFORMATICA

PROTOTYPE SYSTEM FOR ENHANCING HAND NEUROMUSCULAR REHABILITATION EXERCISES EXPLOTING sEMG SIGNALS AND VISUAL FEEDBACK

Tesi di laurea magistrale in Foundations of Industrial Robotics

Relatore	**Presentata da**
Prof. Roberto Meattini	**Filippo Paganelli**

Sessione III dicembre 2023
Anno Accademico 2022/2023

Abstract

The rehabilitation process, after injuries to the wrist or forearm, still consists of repetitive exercises, usually assigned by the therapist together with a poorly detailed explanation on their correct execution. In this way, the patients are frequently brought afar from feeling consistently motivated to complete these exercises.

Technology, instead, is developing fast and all kinds of patients have access to smartphones or computers, nowadays. Physical exercise and activities are now simple to monitor, through health mobile applications and wearable devices, among other solutions.

On these grounds - and as a continuation of a project work carried out at the University of Southern Denmark (SDU), Social Technology Lab 1 course - the RehTracker prototype was built to provide a technological help to the patient, reinforcing the motivation component during their rehabilitation process by taking advantage of visual feedback and amplifying the patient's autonomy.

The physical device was developed following an iterative process building up feature on feature after testing and integration with a mobile application and a website, developed concurrently to work as an interacting ecosystem that serves the user with light-based instant feedback and long-term exercise statistics.

Feedback was collected from physiotherapists to hone the prototype's usability. Then, a user testing phase followed to address the point of view of the hypothetical patient and gather additional observations. Feedback and testing were performed in Denmark, as the chosen case study setting for developing this prototype.

To increase the focus on visual feedback, a robotics visualisation mode was designed and added to provide a real-time representation of the contraction exercises, again increasing the user's motivational gain.

Based on the theory suggestions and the therapists' positive feedback, the author suggests that such a technological solution can be beneficial to the patients undergoing rehabilitation - with an emphasis on wrist/forearm injuries - by increasing the subject's motivation, as it is a key factor in patient's adherence to the rehabilitation process. Patient's consistency is beneficial not only for preventing reoccurring injuries but for saving social welfare resources as well, especially in countries that provide a public free healthcare system.

Preface

This thesis project was carried out at the University of Southern Denmark (SDU) in Odense, Denmark, as a continuation of and improvement on a course project completed at the same institution. This document focuses both on the software components of the project and the hardware ones. However, the physical prototype has been developed with the precious help of Paul Vuong Hung Pham, an MSc student at SDU. The robotics visualisation was realised at the University of Bologna (UNIBO) in Bologna, Italy, at the Laboratory of Automation and Robotics (LAR).

The theoretical background, as well as the grounds on which this prototype is proposed, are also dependent on the country-specific management of the healthcare system. Specifically to this project, the country of reference is Denmark, which provides free public medical assistance to its citizens. The same concept is, thus, also applicable in other similar contexts such as other European countries that provide a similar health assistance.

Contents

List of Acronyms

BLE	Bluetooth Low Energy
DKK	Danish crowns
EMG	Electromyography
FBM	Fogg Behavior Model
GATT	Generic ATTribute Profile
HRI	human-robot interface
IMU	Inertial measurement unit
JWT	JSON Web Tokens
KRAM	kost, rygning, alkohol, motion ([ENG]: diet, smoking, alcohol, exercise)
LAR	Laboratory of Automation and Robotics
NASTRA	Nationale Strategiudvalg for Sundhedsvidenskab ([ENG]: National Strategy Committee for Health Science)
OUH	Odense University Hospital
RC	Radio Control

SDU	University of Southern Denmark
UB Hand IV	University of Bologna Hand, version IV
UNIBO	University of Bologna
VR	virtual reality
WHO	World Health Organization
YLDs	years of life lived with disability

Chapter 1

Introduction

Over the past decades, the health sector has focused on the average life expectancy of Danish citizens and how it is affected by life-threatening diseases. Consequently, political and professional interests and funds have been primarily directed to the prevention of the numerous diseases from which Danish people decease, including cardiovascular diseases, cancer and type 2 diabetes. The main focus has also shifted to the factors that lead to these conditions: kost, rygning, alkohol, motion ([ENG]: diet, smoking, alcohol, exercise) (KRAM) [38].

By contrast, the lifestyle diseases that are not deadly but have an effect on the everyday life have not been focused on to the same degree - e.g. musculoskeletal disorders. Although these diseases do not have a direct impact on life expectancy, they produce very serious consequences for the individual and high direct - and indirect - costs for society [38]. The direct costs include resources consumption in the health sector, while the indirect costs entail a reduced contribution to society due to disability, long-term sickness leaves and absence from the labour market [38]. Moreover, on average, a Danish citizen is estimated to be losing 7 years of good life because of this class of diseases [39].

According to the Nationale Strategiudvalg for Sundhedsvidenskab ([ENG]: National Strategy Committee for Health Science) (NASTRA), musculoskeletal injuries and diseases account for almost half of all occupational diseases, with a socioeconomic cost of more than 17 billion Danish crowns (DKK) per year (DKK in 2010), making it one of the most expensive disease area in Denmark [45]. More recently, it has been estimated to reach and surpass the 20 billion DKK per year (DKK in 2015) - around 15% of the entire diseases-related expenses in the country.

Furthermore, according to World Health Organization (WHO), only around 50% of patients in developed countries - suffering from chronic illnesses - comply to the prescribed therapy regimens. The same statistic is estimated to be even lower in developing countries [54]. The consequence is longer or repeated rehabilitation periods, which contribute to the total costs for society. The repercussions on the individual, instead, can also come in the form of a lower quality of life, when non-complying to the prescribed therapy, by extending the duration of the healing process or increasing the likelihood of future injuries.

Investigating the reasons behind the problem, there seems to be a correlation between patients' non-compliance and some commonly experienced issues: lack of time to exercise, a rehabilitation program non-sufficiently tailored to the patient's specific needs or scarcity of motivation [72].

Motivation, especially, is a very strong element that influences the compliance equation. In this regard, technology can be of great help in boosting motivation - thus increasing the patient's compliance to the exercises. In Denmark specifically, technological solutions have already been - or are on the way to be - proven effective at pushing the patient in the right direction, by allowing for more freedom, in terms of time, and more feedback on the rehabilitation progress [16].

Based on the above findings, this thesis proposes a prototype of such a technological solution, to be looked at as a proof of concept that this is the right approach to tackle the non-compliance issue. A physical prototype, supported by a web and a mobile applications, has been developed with the goal of monitoring the execution of two main classes of exercises, concerning the forearm and the wrist. The final product was named **RehTracker**, as for it aims at tracking the rehabilitation process.

Following an iterative prototype-based process, expert reviews were collected to apply corrections on the final form of the product; successively, user testing was conducted to explore user-experience-related issues.

Exploring the field of research behind the concept of human-robot interface (HRI), a robotic visualisation has subsequently been developed to provide an additional visual feedback mode to the RehTracker proof of concept. It is composed of a virtual representation of a hand which closes and opens accordingly to the data collected through the physical prototype, based on the University of Bologna Hand, version IV (UB Hand IV) [47]. The reason behind providing additional feedback is the possibility to contribute to the motivation boost provided by a technological solution such as RehTracker and point towards possible future implementations of concepts such as exergames and gamification [80].

On the grounds of the theoretical findings and the developed proof of concept, the author's aim within this project is finding out if RehTracker can improve patients' adherence by providing autonomy and visual feedback throughout the rehabilitation process.

Chapter 2

Research and development process

In this chapter, the research methods applied are outlined, followed by an explanation of the development process adopted to transition from the researched theory, alongside experts and user's opinions, to the final version of RehTracker.

2.1 Research and development phases

The research process consisted in six subsequent phases, explained below:

1. **Background:** in this first step of the process, the theory necessary to understand the problem's framework was researched. The main areas of interest were the rehabilitation process and how good/bad do patients comply with it, the reasons behind a discovered low compliance rate, how motivation plays a role in determining the patient's success in performing the prescribed exercises and how can technology help in tackling this issue, by acting on the motivational factors.

2. **Preliminary Expert Reviews:** the first round of expert reviews was conducted to assess the feasibility of a wearable solution to monitor the exercises' execution - while providing feedback to the patient - and to get insights on their direct experience with patients' adherence.

3. **Physical Prototype Development:** based on the results of phases 1. and 2., to further evaluate the feasibility of a light-weight and easy-to-use wearable technological solution, a basic physical prototype was built following an iterative incremental process. The working prototype was then subjected to an expert review to identify eventual defects and strengths.

4. **App Development:** the mobile app was developed following an iterative incremental process, to allow communication with the physical prototype. Then, user reviews were collected to evaluate its user-friendliness.
5. **Website Development:** a web-based mock-up interface was developed to provide an access point for the therapist and for the patients and to set the grounds for future features implementations. This component also hosts the API to communicate with the app.
6. **Robotics Visualisation:** finally, the physical prototype is expanded with the support for communicating with a Simulink model running on a computer.

2.1.1 Background

To be able to understand the knowledge area behind rehabilitation and its limitations, a number of publications were analysed, to begin with. Only a few of them relate to non-recent studies, most are instead very recent evaluations of the rehabilitation compliance issues, but the former were included anyway as they have been backed-up by the more recent ones. Reports from governmental institutions/councils and international organisations were also taken in consideration. The first publications analysed provided an understanding of what rehabilitation means and how it is conducted by professionals. Then, the following publications allowed to understand how motivation plays a role in the process and what can be done to tweak it. Additional papers and the reports provided, instead, a clear idea of the magnitude of the problem - non-compliance to the rehabilitation process - and its costs on the individual and on society. When no relevant publication was found on a topic, web articles were also considered.

The documents analysed have been published in Danish and English. In the former case, partial translations were necessary to allow for a full understanding of the content.

Outcome: this phase made it possible to design the interview questions for the next step, focusing on the main issues found by browsing the documents, to look for confirmations and possible new ideas to explore.

2.1.2 Expert interviews

The interviews were based on a list of questions - and follow-up questions to them - to be able to collect comparable answers from the interviewees, who were selected to be practicing occupational therapists only, two individuals. The reason behind choosing to interview these kind of experts, at this point of the research process, was to be able to compare their answers with the theory found in the previous background phase.

Experts were interviewed to gather their opinion on a few hot points for this project:

- patients' actual compliance to the rehabilitation prescription
- what aids are given to the patient, to help them follow the exercises when they are in a private uncontrolled environment

The replies confirmed that the patients failed to complete their exercise regimen more often when in an advanced stage of their rehabilitation. That is to say when they get less followed by the therapist compared to the beginning of the process.

Outcome: reviewing the experts' answers laid the foundation for the physical prototype. The main focus would have been on ways to provide feedback - as motivation also depends on it. Together with the outcome of phase 1., phase 2. also helped in defining the classes of exercises to focus on.

2.1.3 Physical prototype development

To allow for the maximum flexibility, an iterative prototype-based design process was adopted. Before starting the development process, the technologies were identified in light of wearability and ease-of-use. The technological choices were also guided by the exercises' classes defined at the end of the previous phase.

The prototype went through four main iterations:

1. Implementation of a wearable solution with wireless connectivity.
2. Implementation of a way to detect rotations of the wrist using an inertial chip.
3. Implementation of a muscle contraction sensor and its cables to read the forearm muscles' contractions.

4. Implementation of a light-based immediate feedback system through the use of an LED strip.

Expert review: the same experts interviewed in phase 2. were asked an opinion on the physical prototype at the end of the design iterations. Their responses confirmed the effectiveness of the lights - as a way to provide immediate feedback - and the wearability of the device. They also pointed out ways of improving the user-friendliness.

Outcome: out of phase 3., the main component of the RehTracker was completed and it was time to lay out the other parts and their overall interaction. This phase's decisions on technology narrowed down the research scope for the next phases' choices.

2.1.4 App development

A mobile application was developed to act as the interaction channel between the user (patient) and the physical prototype. As for phase 3., an iterative and incremental design process was adopted. The technological choices were also based on the final physical prototype out of the preceding development phase.

The app went through two subsequent iterations:

1. Implementation of the authentication logic.
2. Implementation of the wireless communication with the physical prototype.

User testing: non-expert testing was conducted to investigate on the ease-of-use and user-friendliness of the conjoint use of physical prototype and app. The users positively highlighted the fact that the physical device is lightweight and easy to wear, while the app helps giving an idea on the progress during the exercise execution. However, more could be implemented in terms of interactivity and issues were pointed out on the use of the muscle pads and on the lights' sensitivity.

Outcome: at the end of this phase, the app was completed as well as the interaction with the physical prototype. Along with the expert review from phase 2., the user feedback reaffirmed the overall correctness of the design choices. The main structure of the next phase was then outlined.

2.1.5 Website development

For the purpose of proposing a complete technological solution - to tackle this project's issue - a mock-up version of a website was realised, followed by the development of the support logic for the app's authentication flow and the future storing/management of the patient's rehabilitation data.

The development process consisted of 3 simultaneous iterations as they led to the implementation of non-interdependent features:

- Implementation of the data storing solution.
- Implementation of the authentication and server/backend logic.
- Implementation of the overview/statistics mock-up page.

Outcome: with the end of the website phase, the technological solution of the RehTracker was completed and ready to provide an exhaustive proof of concept.

2.1.6 Robotics visualisation

Following the development of the physical prototype, the mobile application and the website, a robotics visualisation project was developed. Its aim was to integrate with the physical prototype of RehTracker to provide a real-time visualisation of hand contraction exercises.

Two main phases characterised the development process:

- Establishing the Bluetooth Low Energy (BLE) communication between the physical prototype and the script running on the computer.
- Reproducing the real-time data coming from the wireless connection in a visual form using a 3D model of the UB Hand IV.

2.2 Development process

Developing a project that entails both software and hardware features, especially when they have to work together and interact, requires a very flexible development process that allows for frequent assessments of the current status.

The development methodology adopted for this project is an ad hoc version of Agile development mainly applying Extreme Programming concepts, in a very personalised manner as the work was mostly carried out by a single individual, thus requiring low to null management of people [69]. Moreover, no customer role is identifiable in this project as it aims at developing a proof of concept. For these reasons, the process here described only took inspiration from the already-defined Agile methodologies.

While working on the physical prototype (phase 3.), the approach was time-management-free and the development resources were focused on a continuous integration of features. Each iteration of the process gave birth to a working prototype, which the next feature would have been implemented on, in the following iteration.

The process consisted of the following steps:

1. Identification of the iteration's goals.
2. Research necessary to the specific iteration's goals.
3. Development of the feature based on the outcome of step 1.
4. Evaluation of the result from step 2. and analysis.
5. Conclusion of the current iteration and preparation to the next one, restarting from step 1.

Chapter 3

Background

In this chapter, the findings on the theory behind the RehTracker are explained in detail.

3.1 Musculoskeletal disorders

Musculoskeletal conditions can affect individuals of any age, from children to the elderly. They comprise different areas of the body at the same time: bones, joints and muscles, primarily. These areas can be localised or widespread, and the pain resulting from the condition can be both temporary and chronic.

Among a large number of disease areas, the musculoskeletal one is the biggest contributor to the global need for rehabilitation [fig.: 3.1]. These class of diseases accounts for approximately two thirds of all the cases that would benefit from rehabilitation [21]. It is also of interest to notice that another important contribution to the total number of these cases comes from injuries - considered as a subcategory of musculoskeletal disorders. In the scope of these research process, physical injuries - e.g. fractures - are of particular interest, for they benefit from rehabilitation as part of the therapy treatment.

3.1.1 General costs on society

The numbers on musculoskeletal disorders are not positive in matters of cost analysis. According to the Danish state institute for public health, in Denmark, 8 billion DKK are spent from public money every year, to deal with these disorders [45]. Of these expenses, almost half of it consists in the patient's hospitalisation. Outpatient treatment - i.e. rehabilitation and therapy treatments following hospitalisation -

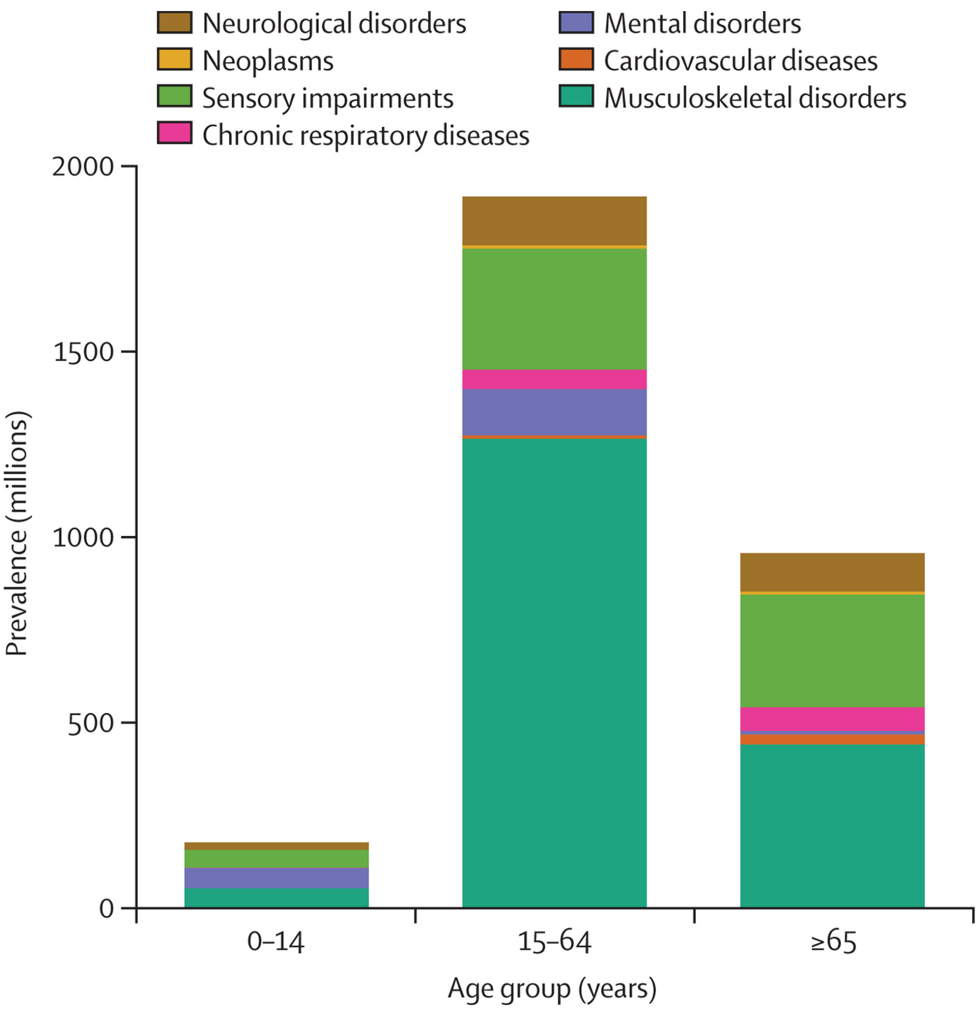


Figure 3.1: Global distribution of the disease areas that would benefit from rehabilitation, per age groups [21].

constitute about one third of it. The remaining is taken up mostly by the cost of the necessary medicines [tab.: 3.3]. Moreover, additional costs arise for the individual, in all the mentioned categories, up to DKK 27,893 per person, for patients that also present activity difficulties.

Beyond these numbers, when pain and reduced mobility are also taken into account, more billion DKK are to be added to the total costs on society [tab.: 3.2].

From this study, run in 2010, the total direct and indirect costs relatable to musculoskeletal disorders can be estimated to be DKK 17.2 billion [45]. In 2015, the

Disease area	Public cost (billion DKK)
Mental disorders	29.068
Musculoskeletal diseases	21.273
Cardiovascular diseases	17.820
Cancer	16.080
Injuries/Accidents	12.371

Table 3.1: The first five most expensive disease areas in Denmark and their calculated public expense, as of 2015.

Patient category	Per person (DKK)	Public cost (billion DKK)
With pain	14,930	9.2
With reduced mobility	27,893	7.2
With both factors	27,158	5.8
With none of them	6,339	8.0

Table 3.2: An overview of the annual extra sick days, per individual and in total among the surveyed individuals. The categories considered are also presenting factors like pain, reduced mobility, both of them or no additional factors beside the musculoskeletal disorder.

Knowledge Council for Prevention in Denmark determined this number to be even higher at about DKK 21 billion, when calculated using the cost-of-illness method, thus confirming a growing trend from the past estimates on these category of public expenses [38]. Additionally, this disease area is reconfirmed to be the second most expensive in Denmark, following mental disorders [tab.: 3.1].

3.1.2 Costs and losses related to sick days

Losses are also visible in terms of production losses and a limited availability of those working people that are affected by musculoskeletal disorders and suffer from reduced mobility and pain, as a consequence of it.

From the report compiled by the **Statens Institut for Folkesundhed**, it is also clear that the extra sick days amount to a number that should not be overlooked [45]. Of those surveyed that were not feeling pain and were not being reduced in their mobility, the number of sick days attributed to their sickness amounts to more than 700 thousand, with a per-person waste of 1.4 days. These numbers grow up to more than 11 sick days per person, when restricting the search to those who were also

Cost contributor	Cost per person (DKK)	Public cost (billion DKK)
Admissions	3,086	3.9
Outpatients	2,036	2.6
Medicine	805	1.0
Other	314	0.5

Table 3.3: An overview of the annual costs - per individual and on the public infrastructure - of musculoskeletal disorders, if not including patients affected by a reduced mobility and pain.

Patient category	Per person	Total amount	Cost (DKK billion)
With pain	3.3	768,719	1.1
With reduced mobility	11.1	479,131	0.7
With both factors	11.3	439,731	0.6
With none of them	1.4	767,370	1.2

Table 3.4: An overview of the annual extra sick days, per individual and in total among the surveyed individuals. The categories considered are also presenting factors like pain, reduced mobility, both of them or no additional factors beside the musculoskeletal disorder.

experiencing high levels of pain while being restricted in their activity [tab.: 3.4].

Sick days also bring in additional expenses - again, both for society and the individual - up to DKK 1.2 billion in the short term.

Calculations on the number of extra sick days - in [45] - were performed using the Human-Capital Approach. This approach sets the basis for the calculations on the employee's point of view and counts any hour not worked as an hour lost [40]. By doing this, it is possible to estimate the value of economic productivity that ends up being lost due to health problems, as well as other non-relevant factors, in the scope of this theory research process [40].

3.2 Rehabilitation

According to WHO, rehabilitation is defined as

> "a set of interventions designed to optimize functioning and reduce disability in individuals with health conditions in interaction with their en-

vironment"

or simply as a help for every person - child to elderly - to gain as much independence as possible, in everyday activities [55]. By doing so, rehabilitation serves the purpose of adapting the environment around the patient to help them overcome any kind of difficulties, varying from motoric to sensory impedance or even eating and thinking difficulties.

There is a number of situations where rehabilitation can be necessary, after a disease/illness period, injury or surgery, and simply because of reduced capabilities caused by ageing [55].

Rehabilitation can be executed in the form of exercises, with or without the help of mechanical/electronic devices or an assisting therapist, in a variety of difference physical contexts. For instance, in addition the usual facilities such as hospitals and ambulatory structures, patients can also be offered the possibility to perform the prescribed exercises from their home, or even at their workplace/school [55].

Besides helping the individual to recover, rehabilitation is also extremely helpful in preventing their future hospitalisations - or deduce the length of such periods - as well as allowing them to return to their jobs and/or private activities, to go back to being active members of society [55].

3.2.1 The need for rehabilitation

The amount of people in the world that would need to undergo rehabilitation, at least once during their life, is higher than it could be expected. In 2019, a study analysed 204 countries, aiming at finding out the reach that rehabilitation's benefits would have if the adequate amount of resources would be employed, by the analysed countries [21].

Globally, turns out that 2.41 billion individuals would benefit from being prescribed rehabilitation treatments during the course of their illness, in 2019. Of those, 1.6 billion are adults aged 15 to 64, with a prevalence of women [21].

Furthermore, the study estimated the portion of potentially healthy life lost - or lived at lower quality - because of such diseases. The number amounted to 310 million years of life lived with disability (YLDs), around 63% higher than the same estimate from a comparable study conducted in 1990 [21].

A number of disease areas were taken into account and musculoskeletal disorders turned out to be the largest contributor to the volume of individuals that would benefit from rehabilitation [fig.: 3.1].

3.2.2 Approaches to rehabilitation

Rehabilitation can be prescribed by a therapist - e.g. physiotherapist, occupational therapist, medical doctor - to be performed either in a supervised environment, such as a hospital, or in an unsupervised fashion at the patient's home, school or workplace. In the scope of this project, the main focus will remain on the latter.

Classic approach: from the Odense University Hospital (OUH), it was possible to obtain an official leaflet used as a guidance/explanation of a selection of exercises, meant to be performed autonomously at home. The exercises' scope is specifically therapy for the wrist, forearm and hands of the patient [fig.: 3.2]. To administer the therapy, this kind of leaflet would be handed to the patient as the only form of instructions. Frequently, a verbal explanation is given as well by the therapist. This old-fashioned but very common approach might not be the best choice in terms of fostering the patient's motivation to comply to the prescribed exercise regimen. However, this point is investigated further down in this chapter.

Technological feedback approach: a more modern approach to unsupervised therapy entails the adoption of technological solutions. For instance, the use of wearable devices can allow for a deeper monitoring of the rehabilitation process [62]. Secondly, it has been proven to be effective in increasing the patient's engagement and motivation, in actively participating in the process - again, this point will be expanded later on in this chapter. Another advantage of choosing technological devices to monitor the patients remotely lies in a possible reduction of the number of individuals necessary to conduct a study that requires data collection (i.e. assessment of treatment efficacy and/or therapy trials), given the precision and reach of the sensors available on the market, as of 2012 [62].

3.2.3 Examples of successful technological solutions

- **Valedo:** developed by Hocoma AG, provides real-time augmented feedback as a response to the user's movements performed using the low-back muscles. The movements are transferred to a game environment to motivate the patient, when doing exercises for low-back pain [62].
- **Stroke Rehabilitation Exerciser:** developed by Philips Research, guides the patient through a series of exercises prescribed by the physiotherapist and

Træn underarm

Drejebevægelse
Hold underarmene vandret og albuerne ind til kroppen.

- Vend håndfladerne opad
- Vend håndfladerne nedad

Træn håndleddet

Læg hånden udover en bordkant.
Underarmen skal hvile på bordet.

- Bøj håndleddet bagover med afslappede, let bøjede fingre
- Bøj håndleddet fremad med afslappede fingre

Træn fingrene

- Stræk fingrene helt ud
- Knyt hånden helt
- Spred fingrene
- Saml fingrene

Figure 3.2: Exercise leaflet for the forearm, the wrist and the hand (from left to right panes) from OUH. The text is written in Danish.

uploaded to a patient unit. A wearable sensor keeps track of the body movements and provides feedback [67].

3.3 Patient's adherence

According to Rand [64], adherence - also called compliance - can be defined as

> "the extent to which a patient's behavior corresponds to the physician's therapeutic recommendations."

More recently, WHO [54] proposed a more complete definition based on the former

> "the extent to which a person's behaviour – taking medication, following a diet, and/or executing lifestyle changes, corresponds with agreed recommendations from a health care provider."

The latter definition has been selected as the reference for this project.

This document will deal with adherence in the form of the patient's compliance to a rehabilitation exercise regimen, prescribed by a physician or physiotherapist, to be completed in a given time frame.

3.3.1 How to estimate adherence

Assessing adherence is not a straight-forward task. The resources available to the therapist consist solely of their previous experience with other patients, the observable therapeutic outcome and their “gut feeling” [64].

From past studies, it appears that this way of measuring patients’ adherence cannot be reliable as most questions - in the therapist-patient interviews - are very simply structured and thus do not allow for an in-depth investigation of the patient’s compliance degree [74]. The result is usually an overestimation of the patient’s adherence [64].

3.3.2 The problem of poor compliance

Given that the extent to which patients seem to be complying to their rehabilitation turns out to be overestimated, considering also the disconcerting numbers on the need for rehabilitation in the world [21], it becomes of relevance to interrogate the numbers again on the problem of poor compliance/adherence.

WHO, between other organisations, published concerning estimates about this issue, in 2003 [54]. In developed countries, slightly more than 50% of patients were found to adhere to the prescribed therapies. This percentage is expected to be even lower in developing countries, especially where access to health care presents more inequalities [54]. It is necessary to clarify that these numbers were estimated only in the realm of chronic diseases, nonetheless, other studies confirm patients’ adherence to be “unsatisfactory” [72]. Thus, this document will consider these numbers as also valid with regards to physical exercise, non-necessarily related to chronic illnesses.

3.3.3 Reasons behind poor compliance

In 1993, a study on the matter of patients’ adherence, investigated some of the reasons behind such a poor compliance - already evident at the time [72].

The interviewed candidates were found to feel discomfort caused by difficult or painful exercises, and to get tired while executing the assigned repetitions. Other common issues can be traced back to a lack of flexibility in scheduling time and place to perform the exercises, resulting thus in the patient’s inability to stick the prescribed regimen, in many cases. Of interest in the scope of this document, are also difficulties related to the lack of feedback. Patients often need to be able to first-hand quantify their progress throughout the rehabilitation process. This element was also found to be missing among the interviewed [72].

In a later stage, further questions were asked about perceived factors that create problems for the patients. Motivation stands out as one of them, together with time and pain [72]. The focus of this project moved on the former.

3.4 Motivation

According to Schunk, motivation is a term that describes the driving force that makes people act purposefully to achieve something. Therefore, it can be regarded as a process by which a purposeful action or activity is initiated and maintained [68].

In practical terms, motivation is generally divided in two types, different in the reasons for which an individual works to reach a goal: **intrinsic** and **extrinsic** motivation [66].

3.4.1 Intrinsic motivation

This type of motivation can be described as related to doing something, because it is inherently interesting or enjoyable to do. Because it is enjoyable, it leads to high-quality learning and creativity, especially in teaching activities. Intrinsic motivation lives in the connection between the individual and the task to be completed by them as it is very specifically dependent on both of them. Important to note is also the lack of external forces that induce one person to perform an intrinsically-motivated task, for they feel naturally driven to do it and can take pauses/resume whenever they prefer [66].

Furthermore, to achieve intrinsic motivation, Ryan and Deci [66] identified three key psychological factors identified as autonomy, competence and relatedness. In line with the purpose of this project, autonomy will be considered as technology can help achieving a higher degree of organisational freedom, associable with the feeling of autonomy.

According to Heidi Lynge Løvschall [16], project manager for digitalisation and welfare technology in Hjørring municipality, Denmark, studies from previously implemented technological solutions for rehabilitation show that more flexibility increases the motivation to rehabilitate by 48%, especially for the working population, as citizens can organise their everyday life as they wish:

> “DA: Det er rigtig svært at tage fri og tage i et træningscenter i det tidsrum, vi har åbent, men borgeren har stadigvæk brug for et målrettet tilrettelagt forløb fra en fysioterapeut, som har ekspertviden. Det kan vi

> give dem på denne måde. Man kan træne, når man har tid, og fysioterapeuten følger med og kan tilrettelægge den rigtige træning for borgeren.
>
> ENG: It is very difficult to take time off and go to a gym during the timeframe, though we are open, the citizen still needs a targeted program from a physiotherapist who has expert knowledge. We can give them that in this way. The citizen can train when they have time, and the physiotherapist follows them and can organise the right training for the citizen."

The studies support the hypothesis that the application of specific technological solutions for rehabilitation provides flexibility in everyday life and thus increases the possibility of self-determination taking place - i.e. autonomy - which is one of the conditions Ryan and Deci argue to be essential for intrinsic motivation [16].

Fogg Behavior Model

The Fogg Behavior Model (FBM) was developed to analyse individuals' behaviours and better understand the relationship between motivation to do an action and the ability to perform it [fig.: 3.3]. FBM states that three factors must converge together for the desired behavior to occur: motivation, ability and trigger - e.g. desire or interest. When a behavior does not occur, it means that at least one of these three elements is missing [35].

In the scope of this project, ability is identified in the individual's flexibility in organising their daily life and in checking up on the correct execution of the exercises, through feedback. This factor is particularly important for this project as one of the commonly experienced issues is doubts on how to perform the exercise correctly and sufficiently [16].

Likewise, both motivation and ability factors must be taken care of, in a trade-off relationship - i.e. if one experiences low motivation, it is necessary that the action is simple to perform, low ability required, before a behaviour change can start taking place [35].

However, the most important factor is the trigger - desire or interest - which must be present before the individual can achieve the desired outcome. Even with high motivation and an easy-to-perform action, the desired outcome cannot happen if the trigger is not present.

Based on the above findings, RehTracker is designed to be used as a proof of concept rehabilitation tool, a framework for increasing inner motivation by introducing flexibility in the everyday life of patients. Using RehTracker, it would be possible to perform the rehabilitation exercises at any time of the day, while monitoring their execution thus contributing to the factors necessary for intrinsic motivation to take place. It is important to remember that the type of motivation the patient experiences is not fixed. It can easily transition from intrinsic to extrinsic motivation and vice-versa. This can especially be the case in scenarios that require a longer processes to achieve success, just as a rehabilitation process typically is.

Figure 3.3: A graphical representation of the FBM: the action threshold represents the line that divides a successful behaviour from a failure.

3.4.2 Extrinsic motivation

Opposed to intrinsic motivation, the extrinsic one is related to activities that are completed for their outcome, rather than for the enjoyment of carrying them out. In general, while growing up, individuals tend to be faced situations - especially socially-demanding ones - that demand extrinsic motivation. As an example, this kind of motivation is responsible for making a student complete an assignment that they wouldn't naturally enjoy to do, but have to hand in to be able to achieve their educational goals, in the long run [66].

Moreover, according to Ryan & Deci's Self-Determination Theory, extrinsic motivation can vary widely to the extent that it is performed automatically:

> "For example, a student who does his homework only because he fears parental sanctions for not doing it is extrinsically motivated because he does the work to achieve the separate outcome of avoiding sanctions. Similarly, a student who does the work because she personally believes it is valuable to her chosen career is also extrinsically motivated because she also does it for its instrumental value rather than because she finds it interesting."

In the above, the first student example concerns compliance with an external control, while the latter student example involves personal approval and an awareness of choice. Though both of these examples represent intentional behavior, their two shades of extrinsic motivation vary greatly in their relative autonomy [66].

Of relevance to this project, two variations of extrinsic motivation will be considered: **external regulation** and **identification**. External regulation involves the classic motivation intended in terms of obtaining rewards or avoiding consequences and/or punishments. Identification, instead, is related to the person identifying with the personal importance of a behavior and thus accepting its regulation as their own.

For the purpose of RehTracker, in relation to **external regulation**, rewards can be delivered via feedback in the form of confirmation, both in real-time and through long-term statistics. According to Anders Stengaard Sørensen, there is a greater indication of commitment to independent rehabilitation when patients can see physical progress in a visual fashion, when it would not be measurable otherwise [16].

Learning goals

Connected to **identification**, Bruce Phillip developed the concepts of game and learning goals, which focus solely on the efforts of the player rather than on skills and abilities. By incorporating feedback, it is possible to provide the player with multiple small success experiences. This can be delivered through visual representations of their efforts, with the aim of improving the player's skills and abilities over time. Bruce Phillip believes that improvement and progress towards the goal are as important as success itself [63].

For instance, a simple way to implement learning goals in games is to give the player feedback, on how much improvement has been made since the previous game. In addition, it is also possible to give feedback on the reason why the game ended, in order to minimise the chances of repeating the same mistakes:

> "Learning goals make people try harder, take more risks, spend more time on a task, become less discouraged when facing setbacks, and, in the end, succeed more frequently." [63]

Based on the concept of learning goals, it can be argued that feedback, in the form of visual representations of improvements and progress over time, can make the individual more motivated to perform the rehabilitation exercises, as it would support the concept of **identification:**

> "Identified regulation involves awarding a conscious value to a behavior in such a way that the action is accepted when it is personally important." [66]

3.5 Visual feedback and exergames

Visual feedback can be provided by using virtual reality (VR) systems as the definition of VR entails the perception to be present and to interact with a virtual environment [22].

Using VR has already been proven to be able to help patients undergoing rehabilitation, by providing a visual feedback on their progress otherwise non-noticeable, especially in early stages of the process when mobility is restricted [18].

A study conducted by Calabrò et al., recorded a higher neuronal activation for those patients that were subjected to the VR assistance while exercising. The suggested takeaway is that providing visual aids/interactions during the rehabilitation activity can help patients recover strength and mobility faster, from a neuronal point of view [18].

Apart from the neurological side of rehabilitation, the effort that the patient is willing to put in performing the exercises has also being suggested to increase in contexts where visual feedback is provided, as investigated by Brewer, Klatzky and Matsuoka [17].

In the scope of RehTracker, these concepts justify and support the implementation of an additional visual feedback exercise mode: the robotics visualisation.

Another key suggestion that visual feedback is worth exploring comes from the concept of exergames. Studies have shown that providing a visual virtual environment to patients, while they perform their exercises, can increase their motivation. This can then turn in better results after the rehabilitation period, as the patient is able to better visualise the movements and feel more motivated to achieve some result in the virtual world provided [80].

Chapter 4

Preliminary expert reviews

This chapter will explain the reasons behind collecting expert reviews before development and the comments that were collected. Starting from the reviews' outcome, it has then been possible to start the prototype design phase, having so re-confirmed that a product such as RehTracker can provide a help to foster motivation and the patient's will to adhere to the prescribed rehabilitation regimen.

4.1 Why expert reviews

Before developing the prototype and conducting the use test on the end users, an expert review was collected. According to the production manager Janet M. Six:

> "Expert reviews are especially useful for finding violations of usability standards and best practices. These are often obvious problems that may or may not cause problems during usability testing. For many of these types of issues, usability testing is not necessary to find them or to confidently say that they are a problem." [71]

A first expert review prior to development was necessary to back up the theoretical framework researched during the background phase, potentially expanding the research area to new ideas. This step aimed at confirming the scarcity of patients' adherence in relation to the common therapist practices.

4.2 Structure of the interviews

As an inspiration, the interviews were set up similarly to a mock user test, in order to collect qualitative feedback. The first few minutes were spent explaining what

RehTracker could be and how it would work after development, both from the patient and the healthcare professional's points of view. Two occupational therapists were selected as the interviewed, both practicing as physiotherapists in an hospital environment or a physiotherapy clinic.

Afterwards, the prepared pre-test questions were asked regarding three main topics [app.: A]. Firstly, the therapists were asked whether they reckoned they're patients follow the prescribed exercise regimens, when they're unsupervised, and why. Then, they were given the possibility to rate their patients' rehabilitation process on a scale from 1 to 7, the latter meaning a perfect/flawless rehabilitation course. Finally, it was asked whether patients are generally given additional aids to guide them through the process, on top of the instructional leaflet discussed in the previous chapter [fig.: 3.2].

4.3 Responses

Addressing the first question, both therapists stated that some patients do not fully comply to the prescribed regimen - as of their knowledge - when not supervised, suggesting that the issue seems to grow worse in later stages of the rehabilitation. The reason behind this might be in the patients thinking their rehabilitation has already completed, ignoring then the prescription's time-frame, or they might be afraid of performing the exercises in the wrong way.

Scores from question 2 range from 5/7 for therapist number 1, to 3-6/7 for the other depending on the gravity of the injury, suggesting that less serious injuries would produce a poorer compliance.

About additional aids, the therapists replied that for the most common injuries it is usual to only supply the patient with the instructional leaflet. Less frequently, indications on useful products to purchase are given.

Outcome

The responses collected from the occupational therapists confirmed the lack of adherence/compliance that part of the patients exhibit frequently, adding some insight in the reasons why this could be the case. The reasons suggested by the physiotherapists correspond to a subset of the ones investigated in 1993 [72].

Additionally, it was confirmed that a leaflet is the main source of guidance for patients, when they are left performing the exercises without the therapist's supervision. Thus, the expert responses were found to be supporting the necessity of a product such as RehTracker, to provide patients with some form of feedback/guidance towards their goal of completing their rehabilitation.

Chapter 5

Physical prototype development

This chapter deals with the incremental phases of the development of the physical prototype of RehTracker. The process will be explained expanding from chapter 2 [par.: 2.1.3].

5.1 Technologies

Before starting to implement some kind of prototype, it was necessary to identify the technologies that would have been the main drivers for the idea to take on a physical form. This pre-step was particularly important as the development freedom would have been greatly affected and restricted by the specific choice of technologies to work with.

5.1.1 Bluetooth

To really embrace the wearable nature of the final device idea, it was chosen to deal with data communication through a wireless Bluetooth connection.

Instead of opting for a Bluetooth Classic solution, it was decided to implement a server-client scenario taking advantage of the more recent BLE protocol. While Bluetooth Classic is optimal for situations that require a large amount of data to be transferred, it also consumes a lot of energy to keep the serial communication open and available all the time [41].

On the contrary, BLE was designed to be low-energy-consuming and thus allows for a longer run time. This feature of BLE made it the best wireless choice for this project, as it would allow for a more compact prototype total size by means of the size of the battery required to run it [41].

It was previously experimented with the Radio Control (RC) technology as well, for it allows for a long-range wireless communication. Though, it was quickly discarded from the options for this project when it became clear that it would have been impossible or impractical to make it work in connection with a smartphone.

BLE architecture

The most common structure of a BLE based system comprises a **Central device**, also called the client, and a **Peripheral device**, also regarded as the server. The former is usually a device with access to high computational, memory and power capabilities - e.g. a smartphone, as it is for RehTracker. The latter is frequently a device with the opposite characteristics as the central one: more suited for data collection or electronic controls than data visualisation and user interaction management - e.g. an embedded device, the physical prototype in RehTracker [19].

GATT services and characteristics: Generic ATTribute Profile (GATT) is the protocol responsible for defining the data exchange in the BLE connection-oriented paradigm.
The peripheral device generally advertises services and for each of them their available characteristics. It is also useful to take a look at the characteristic properties. In the domain of this project, only three properties will be explored:

- **Read:** when enabled, the client device can read the characteristic's value.
- **Write:** when enabled, the client can write to the characteristic's value, expecting an acknowledgement response from the server.
- **Notify:** when enabled, starts a server-initiated data transfer, meaning that the client will receive value updates for the specific characteristic directly from the server.

A client device can perform a number of operations. First, it will usually discover the available services and the characteristics they advertise. Then, it will have the option to perform a read operation or a write operation, from and to a characteristic value. Finally, it will be able to negotiate connection parameters such as the packet size and others. For a better understanding, refer to the example scheme [fig.: 5.1].

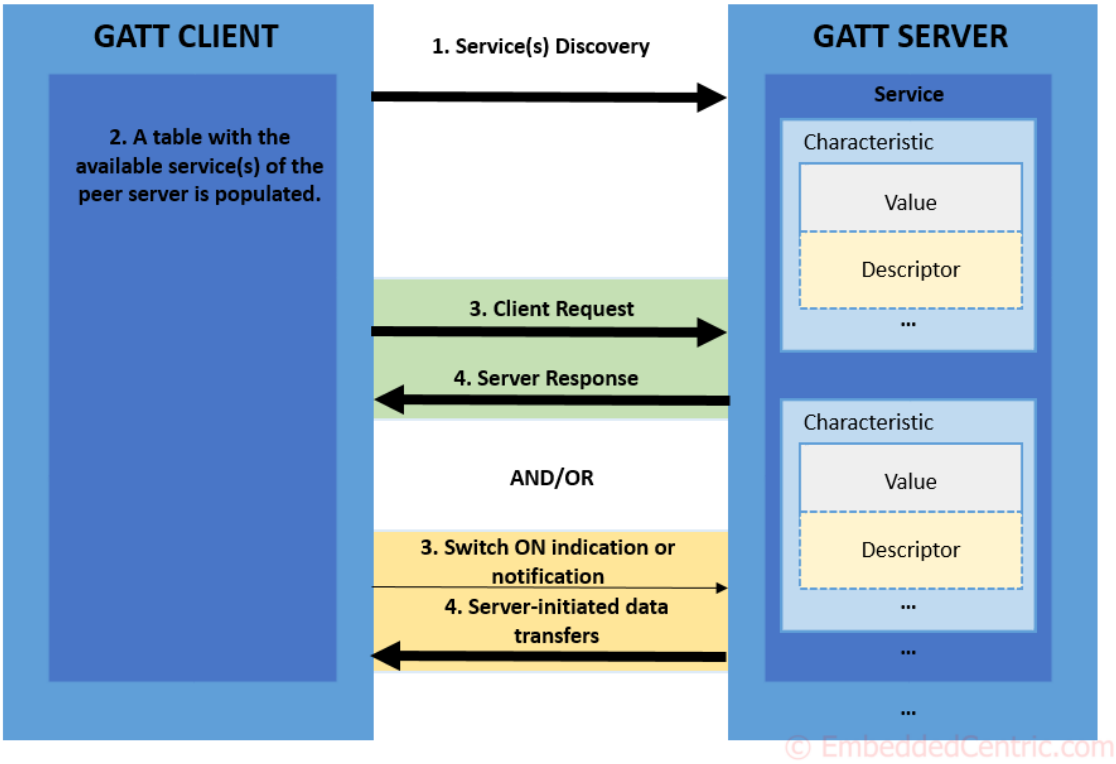


Figure 5.1: Example scheme of a GATT server-client interaction flow [19].

5.1.2 Muscle sensor

Reading the contractions of the forearm muscles requires a muscle sensor capable of analog reading and outputting, in a portable fashion, again, for the sake of creating an easy-to-use portable device in its entirety.

The **MyoWare Muscle Sensor** was chosen as it is a small enough sensor to be incorporated in the physical prototype and its documentation is extremely clear on how to use it [1]. Its integration with a simple microcontroller platform would have been straightforward, as well as the application of the sensor on the patient's forearm.

Sensor structure

Electromyography (EMG) generally is a diagnostic test performed by recording a muscle's - or a group of muscles' - electrical activity by placing electrodes on the portion of skin immediately adjacent to the muscle. Then, an electronic board measures and interprets changes in the electrical potential of the skin, while the muscle contracts, or when it is at rest.

The MyoWare sensor picked for this project has two electrodes onboard ready to be placed on the skin area of interest [fig.: 5.2]. In this scope, it was later opted for the use of independent electrodes with cables, for a cleaner signal reading and better portability of the device [fig.: 5.4]. In order to use the cables, it was also purchased a cable shield. It only serves the purpose to extend the signal connection to a jack connector, which accepts the cables connector and passes the data to the main body of the sensor [fig.: 5.3].

To read data from the sensor and provide power to it, three connection points are available on its board. Two are for power supply (V+ and GND) and the last one is for analog data output (SIG point), see the datasheet for further details [1].

Finally, the last element that is crucial for the sensor to work properly is the surface EMG pads, to be applied to the electrode heads for contact with the skin [fig.: 5.5].

Figure 5.2: Lower side of the muscle sensor, with the onboard ready-to-use electrodes [30].

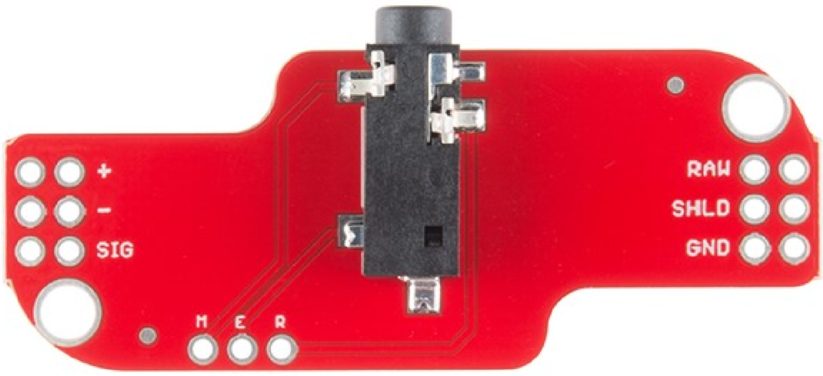


Figure 5.3: Cable shield to connect to the cables via the jack connector and to the myosensor to extend the electrodes reach [29].

Forearm muscles

The important muscles, in the scope of the exercises monitorable with the ReHTracker, are the **Flexor Digitorum Profundus Muscle**, the **Flexor Digitorum Superficialis Muscle** - responsible for flexion of the four fingers excluded the thumb - the **Flexor Pollicis Longus Muscle** - takes care of the thumb flexion towards the palm - and **Pronator/Supinator muscles** that carry out the wrist rotation [53] (see schemes [fig.: 5.6]).

Given the position of these relevant muscles, it is important to note that the electrodes - together with the electrode pads - are to be installed on the part of the skin of the forearm midway between the elbow and the middle of the forearm. Moreover, because the focus is not on the extensor muscles, the relevant side of the forearm is the **anterior compartment** [20].

5.1.3 LED strip

As part of the motivation fostering, it was decided to utilise an immediate visual feedback element. Specifically, keeping in mind again the ease of integration with microcontroller systems like Arduino, the choice fell on an RGB multi-colour 8-LED strip from Adafruit [31].

The strip is very easy to program and provides full control over the eight LEDs, together with full support for a large variety of colours, thus allowing for a more creative visual feedback design than just an on-off light configuration. Additionally,

Figure 5.4: External cables with electrode functionality, to be plugged in the main body of the sensor through an AUX interface [32].

the size of it was considered to be perfectly in line with the portability requirement set for the project [fig.: 5.7].

5.1.4 Arduino Nano

After a number of tries, the brain of the physical prototype was identified in the Arduino Nano family of devices. They are open-source microcontroller boards equipped with digital and analog input and output pins, for designing interactions with a vast amount of external modules/devices. Moreover, they are extremely easy to program given one's knowledge of the programming language C++, of which Arduino boards support a variation [70]. The users community - together with embedded systems/boards/electronics companies - create and maintain software libraries to integrate new hardware and software components, continuously.

Differently then the Arduino basic family - e.g. the Arduino UNO board [6] - the

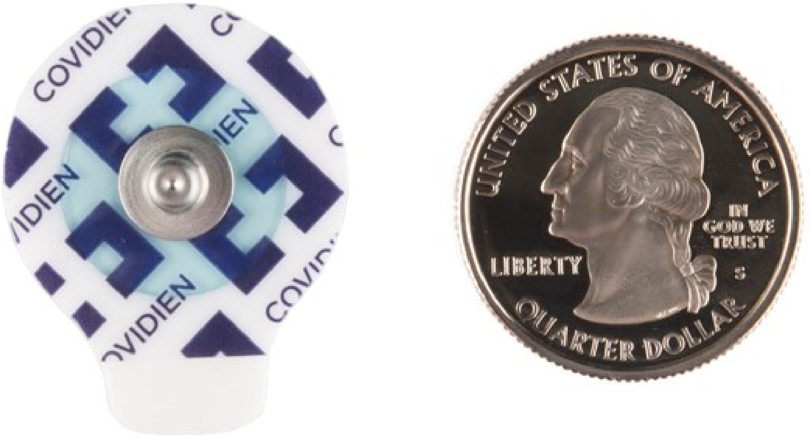


Figure 5.5: An EMG pad, compared side by side with a quarter-dollar coin [28].

Nano boards are much smaller in size and compact practically the same features in a tighter board [44].

Arduino Nano 33 BLE

On the grounds of the previous technological choices, the **Arduino Nano 33 BLE** was selected as the ideal microcontroller board for managing all the connections and data exchange for this prototype [4].

In addition to the possibility of implementing BLE native solutions, thanks to the onboard Bluetooth module, it also embeds an Inertial measurement unit (IMU) for measuring 3-axis acceleration and more. This was identified as the best component for measuring wrist-rotation events, thus enabling the recording of those kind of exercises [fig.: 5.8].

On the matter of portability and ease-of-use, alongside the compact shape and the minimal size, it uses a Micro-USB port, taking a smaller space than a USB-B one as in Arduino UNOs [fig.: 5.8].

5.2 Development iterations

Having identified the right technologies for building a prototype, it was time to start assembling the parts together. Following, is a breakdown of the main iterations that characterised the development of the physical part of RehTracker.

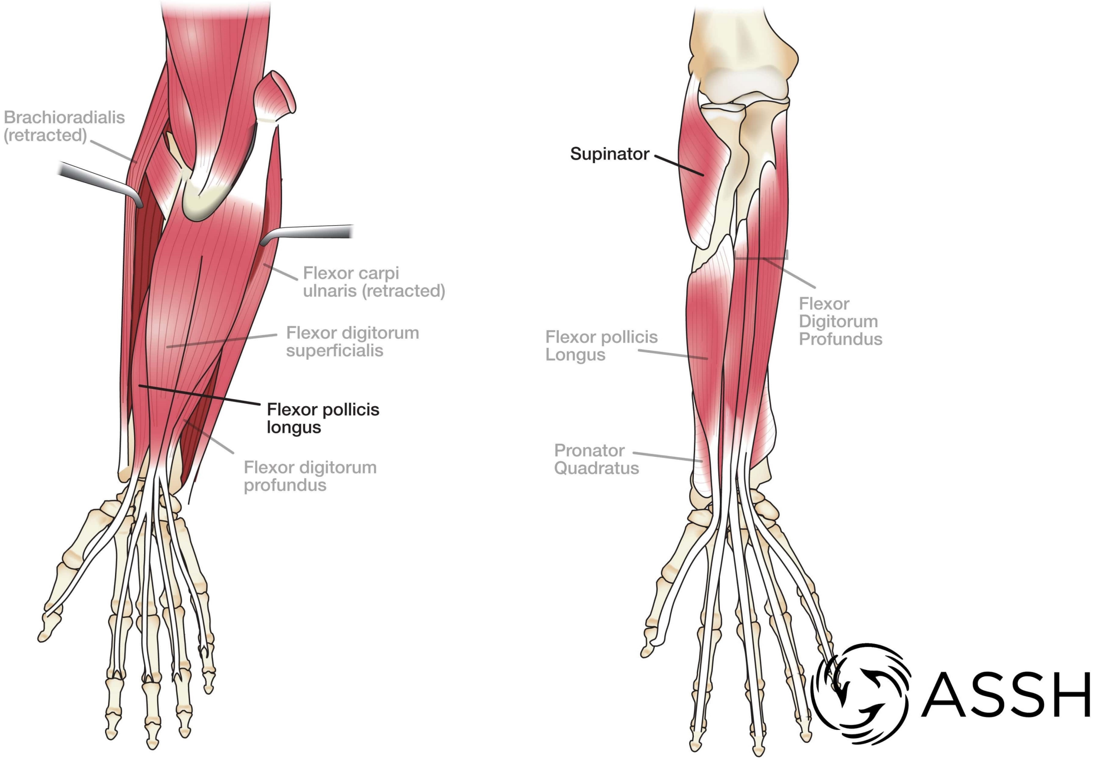


Figure 5.6: Left: a representation of the main flexors that produce the closing movements of the hand fingers. Right: a representation of some of the muscles responsible for the wrist rotations [76].

Throughout the entire process, the software **Arduino IDE 2** was used to write the code for the board and test it directly on the board itself [9]. The process consists of writing a snippet of valid Arduino code, compile it using the Compile feature of the software and upload it to the Arduino board - connected to the computer via a USB connector.

Regarding the code explanations, only important functions and parts of it will be mentioned as the full code script is easier to read on GitHub, where it is entirely accessible [56]. Refer to the repository for the full code.

General code structure

An Arduino code script, also referred as **sketch**, is structured in three main parts:

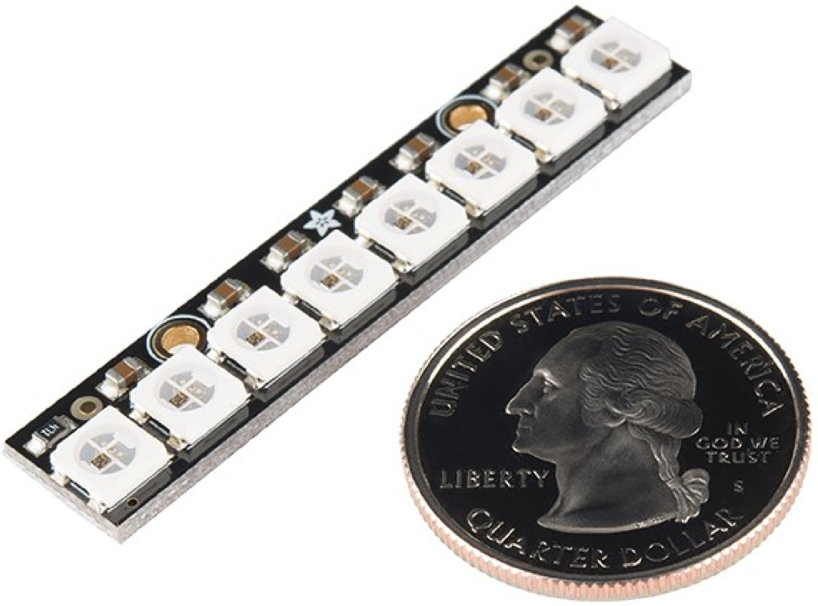


Figure 5.7: LED strip size comparison with a quarter-dollar coin.

- **initialisation/definition:** in this first part of the script, it us usual to initialise - or simply define - some variables/objects to be used later on.
- **setup:** the code contained in this function is run only once at the board's startup or reset. It is commonly used to initialise variables or start using libraries [11].
- **loop:** this function is run continuously in cycles. Here goes the main business logic of the entire sketch, to be designed accordingly to the looping nature of this function [10].

5.2.1 Iteration 1. Arduino and BLE

This first iteration aimed at setting up the Bluetooth functionality of the Arduino Nano 33 BLE board.

In order to control the Bluetooth module and implement the peripheral role of the GATT paradigm (GATT server in the scheme above [fig.: 5.1]), the **ArduinoBLE** library was installed as it is the reference for enabling the communication functionality

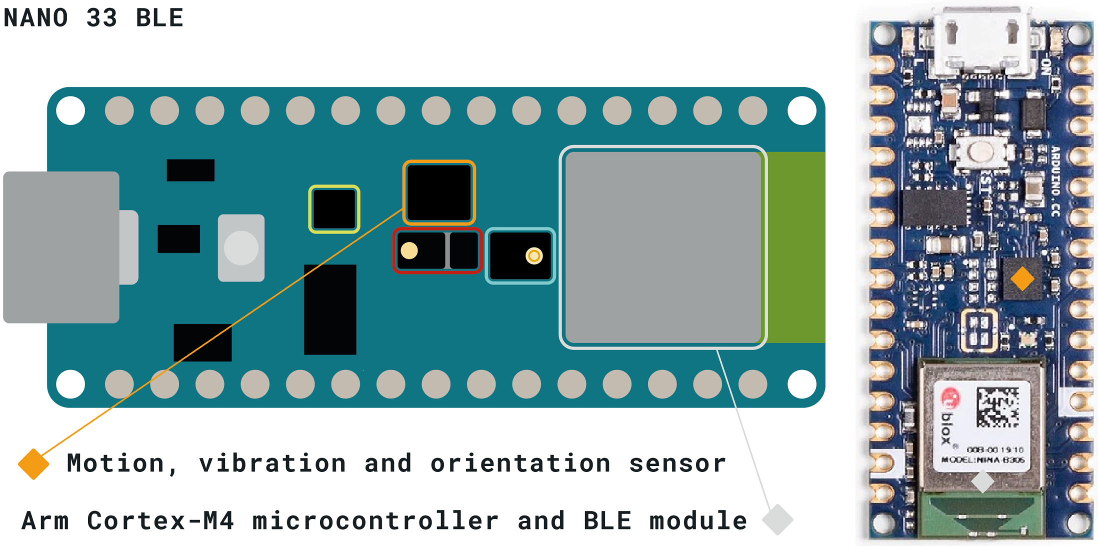


Figure 5.8: Left: a scheme of the high-level components install on the board that are relevant to this project [5]. In orange the IMU, in grey the BLE module. Right: a top-view picture of the board [4].

on the board of choice and many others [8].

Initial definitions

To begin with, a **BLEService** instance is defined as the advertised service that wraps the necessary characteristics for the Arduino-mobile app interaction to work. This part of the process did not need any particular care, as well as the following definitions, because it would have only laid out the basic structure of the interaction, leaving out specific details for later.

Following, two **BLECharacteristic** instances are created: one carrying an integer number value that will represent an exercise counter. The count to track would be stored in the same characteristic value and re-utilised, both for summing the number of valid muscle contractions and for counting the valid wrist rotations. This choice was made to optimise the resources used by the code when running on the Arduino board.
The second characteristic was designated to keep track of the type of exercise being recorded at the specific time of check of its value, from the mobile app. Thus, it is defined to be of type boolean following the correspondence **muscle contraction**

exercise = true and **wrist rotation exercise = false**.
Note: the wrist exercise is also referred to as **gyro exercise** in the codebase as it counts on readings from the IMU on the board, which includes a gyroscopic sensor [56]. This part of the code was only written in preparation for the following iterations as the main concern of this iterative phase was to establish the Bluetooth communication channel.

Setup code

In the setup function, the BLE module is initialised. First of all, the device's local name is set to **RehTracker - Console**, this will be a reference when describing the mobile app. Then, the BLEService is registered as the one to be advertised to central devices, when they are scanning for services. However, prior to this setup code, the characteristics are initialised to their starting values: 0 for the counter and true for the exercise type (defaults to the muscle contraction exercises).
Note: the choice of the device's name was made purely for the purpose of being able to identify the BLE device from the app scanning code.

Loop code

As the looping code, a **BLEDevice** is created to become the central one (client) upon successful connection. Then, if and while a central device is connected, the actual logic of the exercise counts is implemented, together with the necessary code handling for when the exercise type is changed externally by the central device.

Testing

To make sure the code was working as intended and was ready for the next iterations, a BLE testing app called **nRF Connect for Mobile** was used [12]. Installed on a mobile device, it allows to discover advertised services and related characteristics, as well as their value types. By using this app, it was possible to apply the necessary modifications to the code to get the Arduino board to work properly. It was then time to proceed with the integration of the other technologies.

5.2.2 Iteration 2. Arduino and the IMU

As the first iteration completed, the focus shifted in integrating the code sketch with the possibility to detect a rotation of the wrist. To start, it was decided to set as a guideline the final position of the device on the user's forearm, to then try to solve the problem of how to detect the right movement.

The position was selected to be right before the beginning of the hand, to be worn on the right edge of the external side of the wrist - the one that is facing the same side as the dorsal side of the hand. Additionally, the Arduino board was aligned with its longer dimension to run parallel to the forearm when the hand is at rest [fig.: 5.9].

In order to find out the proper way to detect rotations, the **Arduino_LSM9DS1** library was chosen. It has been specifically designed for managing communication to/from the LSM9DS1 IMU soldered onboard [7]. From this library it is possible to read three-dimensional information about acceleration, angular rates and magnetic fields.

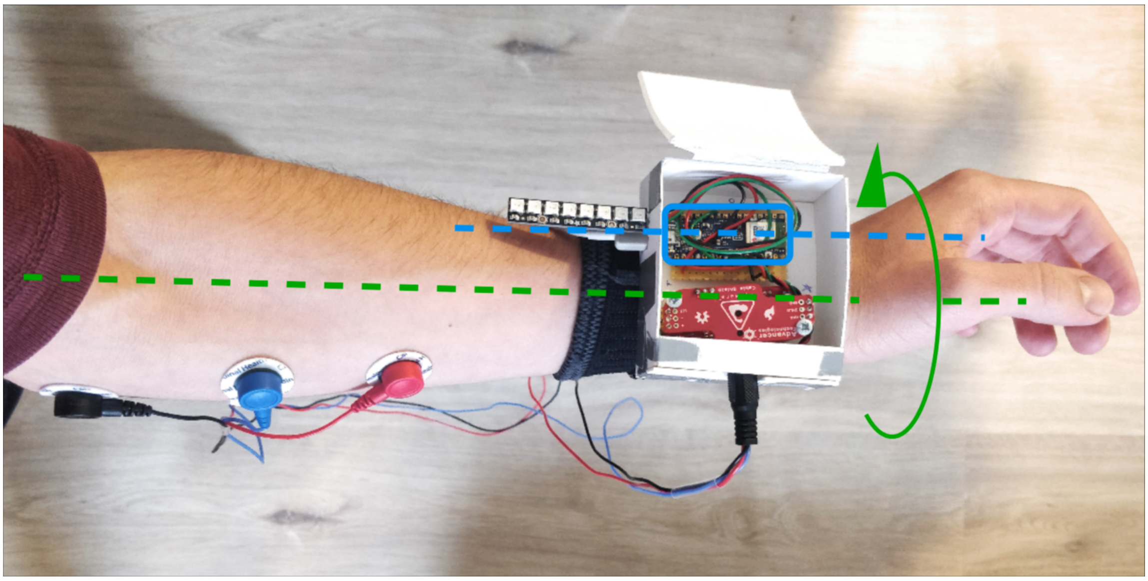


Figure 5.9: A scheme showing the Arduino alignment on the wrist. In green: the rotation axis of the forearm. In blue: the rotation axis chosen for the Arduino, parallel to the one of the forearm. The board is highlighted by a blue rectangle.

After thorough testing with the accelerometer readings from the built-in sensor of the board, it was concluded that - again, given the chosen alignment - the component to look at was the degree angle on the y-axis [2]. That is to say, the reading of a change in the angle's sign would have to be interpreted as a completed half-rotation. For example, starting at the initial position shown in the scheme [fig.: 5.9], rotating

the forearm to one side then coming back to the starting point and rotating to the other side, would have been interpreted as a completed half-rotation.
Note: rotations in both directions must be acute more than a specific threshold to be recorded. This is important to avoid counting rotations when natural unstable back-and-forth movements occur.

Initial definitions

Just like it is for the Bluetooth code, the necessary variables are defined as **ax, ay, az**, representing the acceleration angles for all the axis: x-axis, y-axis and z-axis, respectively. Despite the fact that, as mentioned earlier, only the y component is of interest for the actual rotation counting logic, it is still necessary to read from all the available axis when calling IMU's **readAcceleration** function; **ax** and **az** are simply ignored throughout the sketch.

Further down the script, a separate function is defined to handle this type of exercises by the name of **handleGyroExercise** [56]. Here, the angle variables are read and the counter is updated when a rotation is completed.

Setup code

In this block, the IMU module is initialised, integrating the code with the BLE lines written in the previous iteration.

Loop code

In accordance to the code that selects the correct handling logic based on the current exercise type specified - refer to iteration 1. - the rotation counting function is called to delegate the exercise handling.

Testing

Ensuring that this setup was working correctly was a straightforward process. It consisted of simply recording the angle with the device worn as shown in the scheme [fig.: 5.9], while looking at a testing print of the rotations counter to verify they were counted as expected. Next, it was time to implement the muscle sensor and a new iteration started.

5.2.3 Iteration 3. Arduino and the myosensor

Hereafter, the implementation of the myosensor will be explained, first from a code point of view, then from the electrical connections and physical installation perspective.

In order to initiate a communication with the MyoWare muscle sensor described in the technology overview - see 5.1.2 - no particular library is necessary. The sensor is ready to be used right after a very short delay, following its start up.

Initial definitions

As mentioned earlier, no definitions are needed to read from the myosensor. However, to write a clearer code, it was decided to define the analog input pin **A0** provided by default by the board as **EMG_SENSOR_FRONT**. A few thresholds and useful constants are also defined in this part of the sketch - e.g. the maximum contraction level to consider the repetition a complete contraction.

It is defined, again, a function for handling this type of exercise in order to keep the code readable and as modular as possible. The function is named **handleMuscleExercise** and has the responsibility of incrementing the exercise counter every time the contraction level surpasses a specific threshold [56].

Setup code

No setup code and no initialisations are defined as the communication channel is already established, through the physical connector to the analog input pin.

Loop code

In accordance to the code that selects the correct handling logic based on the current exercise type specified - refer to iteration 1. - the muscle contraction counting function is called to delegate the exercise handling.

Testing

For this iteration, the final test phase represented a more consistent challenge than the previous ones. A reason for this is the process of creating a physical connection between the sensor and the board, which produced not very precise readings at times. Secondly, the inexperience with electrical systems and signal acquisition made troubleshooting longer than expected. For example, it was realised after numerous tries that the board had to powered up without grounding and from a stable DC source, to minimise signal distortions from the myosensor.

Eventually, it was possible to place three pads on the right spot of the forearm and fine-tune the reading constants to produce the expected contraction counting.

More about the wiring and physical connections will be explained further on.

Physical connections

The myosensor was connected to the 5V power pin available on the Arduino board, together with the ground pins, using jumper wires - in red and black in the scheme, respectively [fig.: 5.10]. Then, the cable shield was connected on top of the sensor to be able to use the cables, which would be plugged-in from the jack connector on the shield itself. For the purpose of testing, the pads were applied to the electrodes and placed on the skin as shown in figure 5.9 (the three electrodes are black, blue and red, placed close to the elbow joint).

5.2.4 Iteration 4. Arduino and the LED strip

The last component to be added to the physical prototype was the LED strip, again, to provide immediate feedback to the user. The idea was to display the intensity of the contraction/rotation mapping it linearly from the interval [minimum threshold, maximum value] to a numeric integer interval starting from 1 and growing to 8. This interval would be then associated to the number of LEDs to turn on, on the strip, generating a total of eight possible states to be shown. See scheme for a visual explanation [fig.: 5.11].
For a better visual appeal, it was chosen to light the LEDs up with colours: red, yellow, green and blue.

To control the LED strip, the **Adafruit NeoPixel** library provided the necessary definitions to communicate via I2C protocol to the strip [3].

Initial definitions

Included the library, a **Adafruit_NeoPixel** instance is created. Then, an array with the colour instances is defined.

Just like for the previous iteration, to keep the code clean and easy-to-read, separate functions are defined to calculate the right colours to show at every loop and to actually write them to the strip [56].

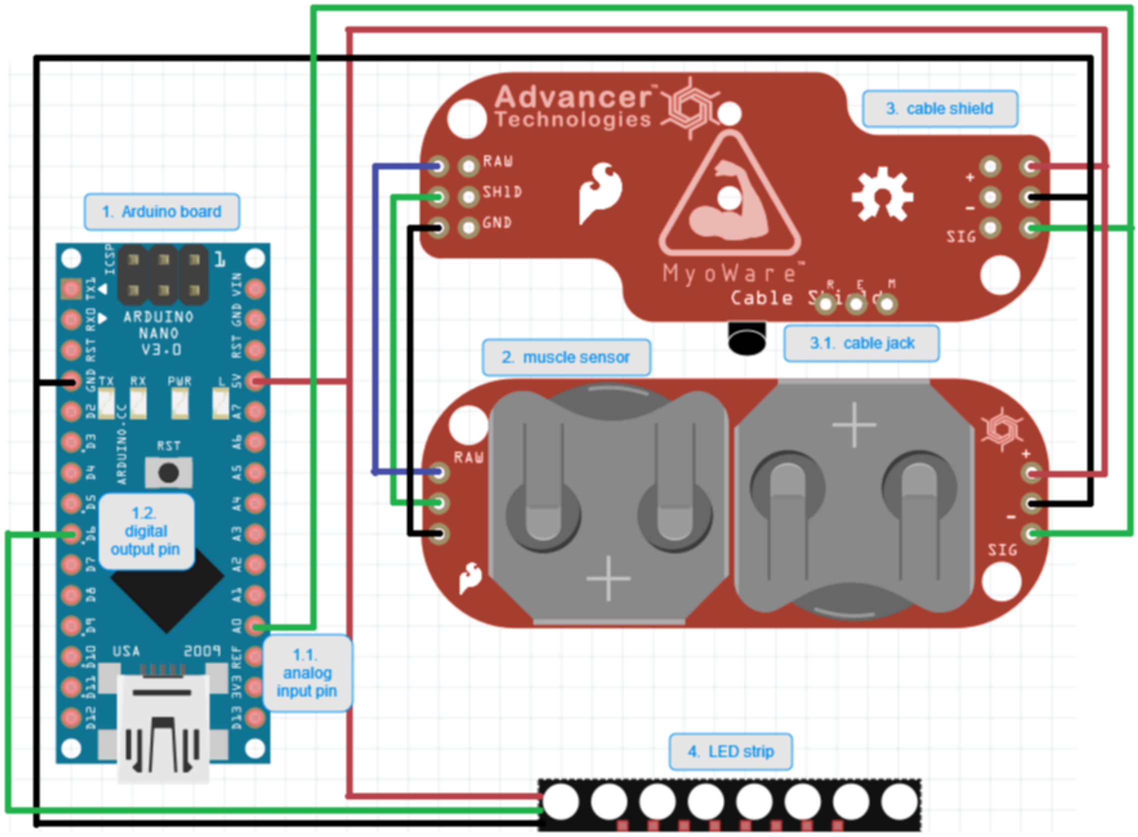


Figure 5.10: A scheme showing the electrical wiring of the physical components together. In red: 5V power lines. In black: ground lines. In purple: raw signal analog line from the cable shield to the muscle sensor. In green: data lines, from the LED strip to the board and from the myosensor to the board.
1.1.: analog input pin A0 for reading the muscle contraction from the sensor. 1.2.: digital output pin D6 for writing to the LED strip. 3.1.: input jack for the external cables to the electrodes.

Setup code

To setup the strip, the connection is first initialised, then the colour is set to be black - represented by the LEDs being turned off - for all the LEDs.

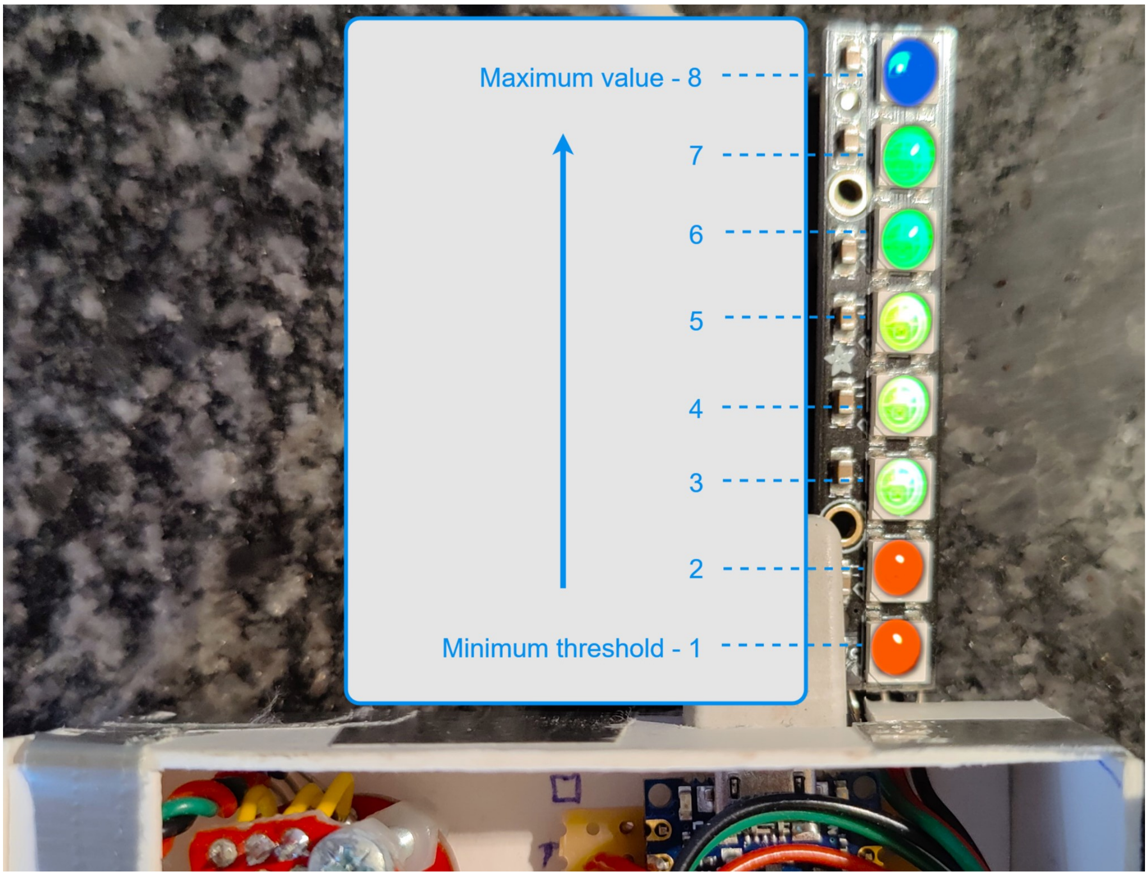


Figure 5.11: A visual mapping of the LEDs when turned on with the corresponding number values used in the Arduino code, from the minimum threshold to the maximum value. In the picture, the strip is representing a state of maximum contraction/rotation level.

Loop code

Every time a loop cycle comes to an end, the functions mentioned before are called to update the LEDs and show the correct colours based on the intensity level of the exercise.

Testing

The LED strip was tested as part of the entire prototype by simulating a full rotation exercise, followed by a contraction exercise. Adjustments were made to the brightness level and to the end-of-loop delay to find the best balance between real-time

LED feedback and the actual limited computational capabilities of the board, when connected to all the components.

5.3 Final prototype

Ending the prototype phase, each iteration turned out to have added an important component to the final idea, which reflected the goals set at the beginning of the process.
To remind them briefly:

- **portability:** the final form of the device created is light in weight, making it easily portable. Also, portability is re-enforced by its wireless nature given by the choice to go for BLE.
- **immediate feedback:** the LED strip addition provides a colourful real-time feedback.
- **compactness:** overall, even after adding all the components and taking care of the wiring/connections, the outer casing takes up a very small space to be a proof of concept.

In virtue of compactness and portability specifically, a simple cardboard casing was assembled, the idea behind being to hide and protect from touching the main board, the wiring and the muscle sensor. Out of the box, protrude only the LED strip - which has to be visible the patient while exercising - the EMG cables and a cable eventually connecting to the board through its micro-USB port [fig.: 5.12].

Therefore, a cloth wristband was purchased with the idea of using a Velcro joint to easily wear the device on the forearm as shown in figure 5.12.

About the final code sketch, it puts together the business logic necessary to make all the components work in harmony. The flow chart shows how the code works at a high-level, see figure 5.13.

5.4 Expert reviews

To target potential flaws of the physical prototype, the same experts - i.e. therapists - interviewed before development were asked to express their opinion a second time

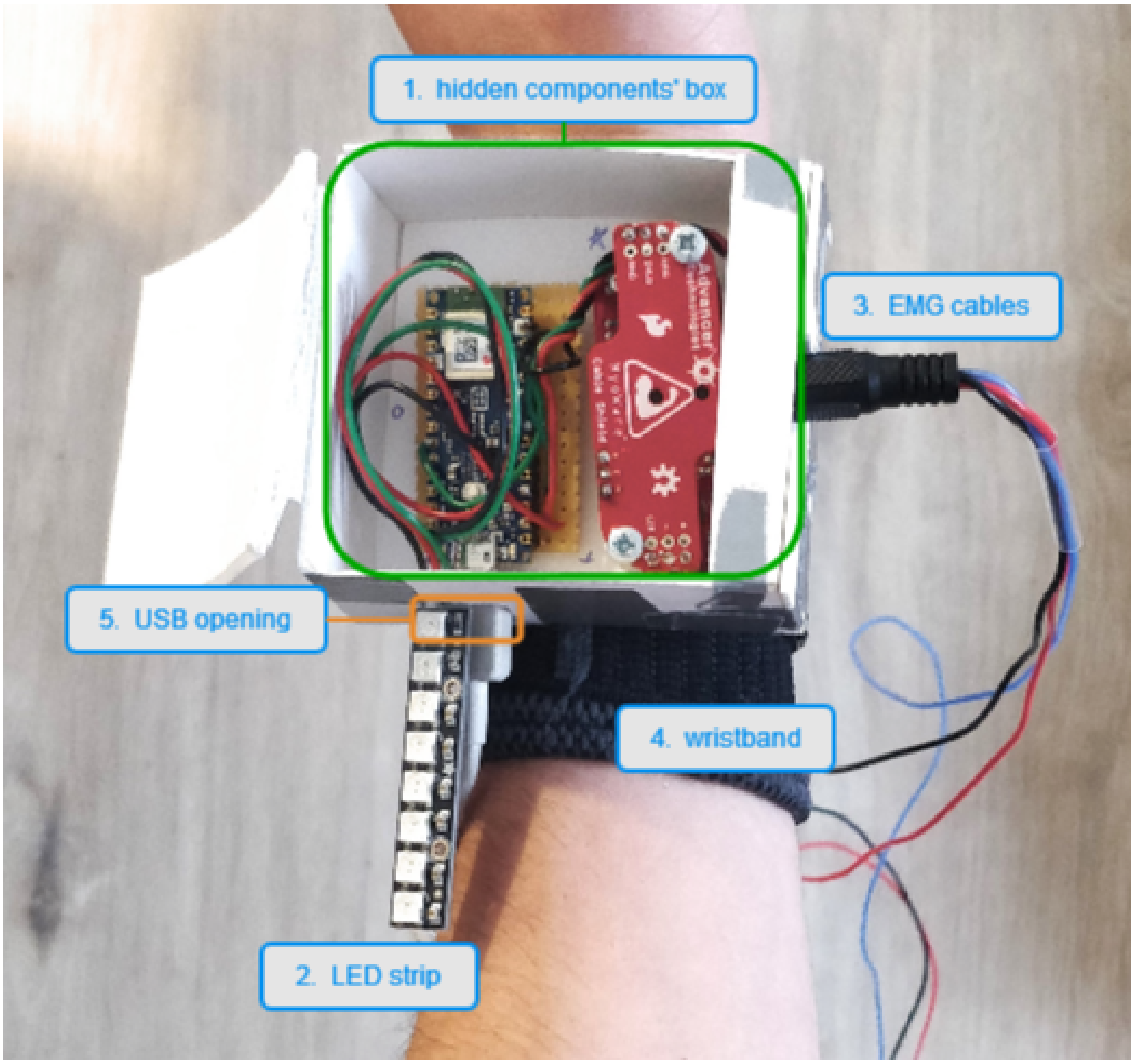


Figure 5.12: A scheme showing the final prototype worn on the wrist. 1.: the cardboard box to contain the main board and the myosensor. 2.: the LED strip sticking out from the box. 3.: the EMG cables exiting the box, connected through the input jack in black. 4.: the cloth wristband to strap on the box to. 5.: the opening for the USB connection, placed underneath the opening for the LEDs.

[par.: 4]. The aim of these late expert reviews was to collect the experts' feedback on the prototype to use as a guideline for future improvements or augmentations.

5.4.1 Structure of the interviews

At first, the prototype features were explained in detail. Then, instructions were given on how the wearable was designed to be put on.

Later, the prepared questions were asked to the therapists [app.: B]. To begin with, the therapists were asked their opinion about the feeling of wearing the prototype on their wrists, the ease of wearing and the lights as a real-time visual feedback mechanism. Then, to confirm that it was suitable as a rehabilitation aid, they were asked their opinion about the classes of exercises targeted by RehTracker. Finally, the questions were directed towards the possibility of using the device as a real rehabilitation tracker, as well as with their patients, potentially.

5.4.2 Responses

To the first three questions, both replies point to the light weight of the physical prototype as a good quality, confirming the high wearability of the device. However, both therapists pointed out that LEDs felt very/too sensitive at representing the contraction level. After they were asked to put the device on by themselves, two things were pointed out. Therapist number one reckoned that the placement of the device on the wrist could be facilitated by addition of symbols to guide the patient. The second therapist mentioned that a patient could encounter difficulties in placing the muscle pads and snapping the wire electrodes on them.

On the question about the exercises, they both agreed on the possibility to cover more exercises and muscles in the future.

About the potential of RehTracker's physical device and future use scenarios, they underlined the real-case application of monitoring the patient's progress and the effectiveness of providing feedback to the patient. One of the therapists also would agree to let some of their patients use the device.

Outcome

Overall, the wearability feature of RehTracker's physical component was recognised successfully, especially in terms of light weight. In addition, these expert reviews proved the effectiveness of the LEDs as a form of visual feedback.

However, the lights will have to be re-calibrated to ensure that the sensibility level is appropriate to the contraction intensity, issue pointed out by the subjects interviewed. Another focus point for the future is making the pads easier to connect to the cables or provide the patient with instructions on how to put them on.

The exercises used a reference for developing the RehTracker have been confirmed to be relevant and the therapists hinted at the possibility to expand the exercises set, pointing at the device's flexibility in terms of rehabilitation coverage.

Finally, the prototype together with the final ecosystem were suggested to be potentially useful in a real-scenario application.

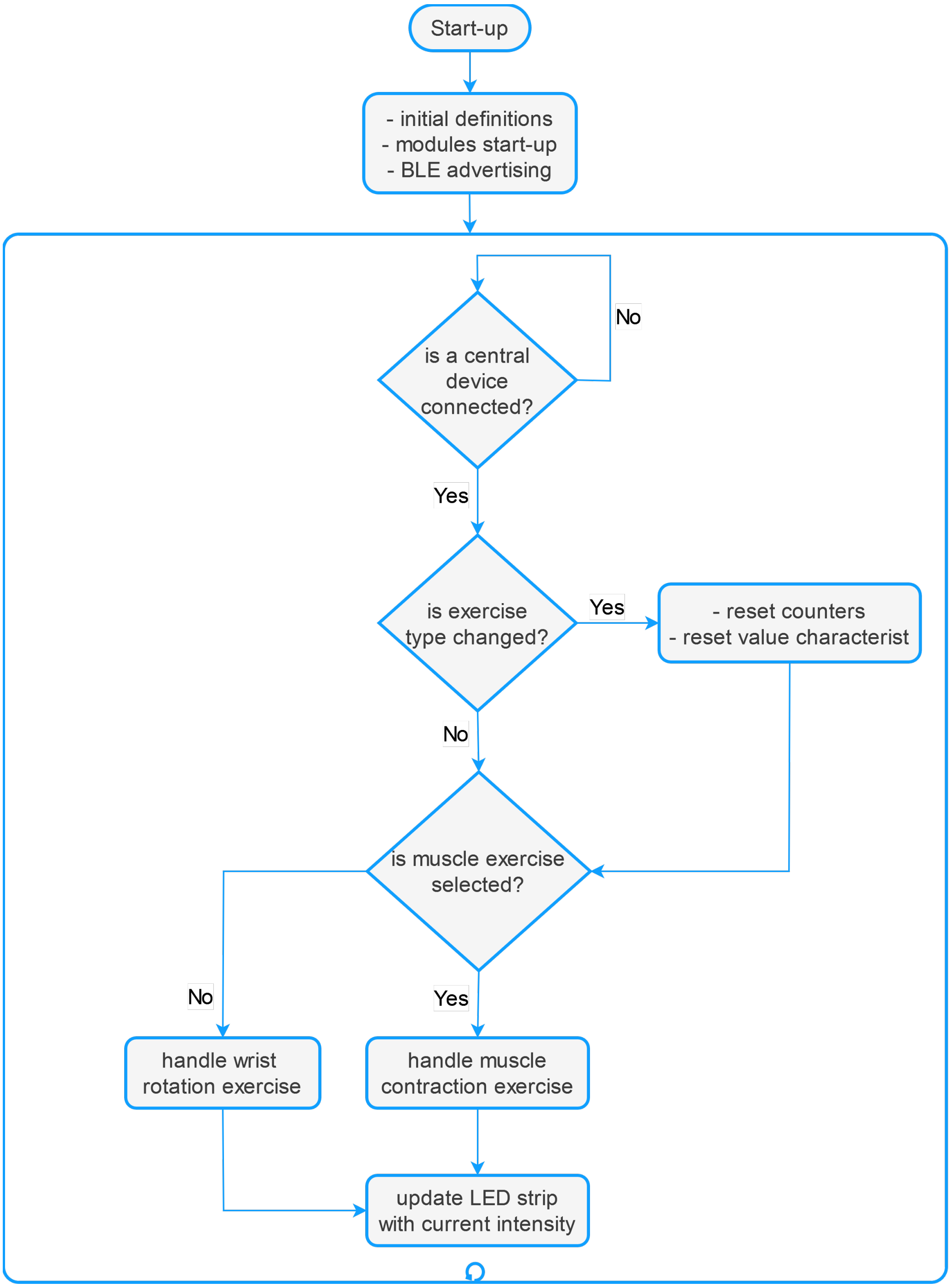


Figure 5.13: A flow chart showing how the final code for the Arduino board works, at a high-level. The last block from the top represents the loop section of the code script.

Chapter 6

Mobile app development

This chapter deals with the incremental phases of the development of the mobile app of RehTracker. The process explained expands from chapter 2 [par.: 2.1.4].

6.1 Technologies

Similarly to the process that preceded the development of the physical prototype, the right technologies had to be identified for the mobile app, before starting with the development incremental iterations. Thus, following is an explanation of the technological choices made for RehTracker's mobile application.

6.1.1 Flutter

Among the available frameworks for mobile app development, a handful of those are called cross-platform. The term refers to their characteristic to allow to share a fair share of the codebase for a mobile app between more than one target platforms, to develop for.

A popular example is **React Native**, a framework based on the idea of sharing the UI components from **React** across multiple platforms such as Android, iOS, Windows, macOS and more, while also allowing to fine-tune all the desired details according to the specific platforms individually.

The clearest advantage of this approach is the possibility of writing most of the application's code only once for all the development targets, saving time during the project's development.

For this project, **Flutter** was chosen, which is another very popular and highly maintained cross-platform development framework [34]. It comes with its own strongly typed programming language called **Dart**, which is highly UI-oriented and very well assisted by plugins for a number of programming environments [33]. In the case of this project, the entire app code was developed using **Visual Studio Code** and the suggested Flutter/Dart plugins to help in the code editing [49].

On the reasons behind choosing Flutter instead of React Native or other options, it was mainly a matter of the performance level that is possible to achieve with Flutter, rather than with the other cross-platform frameworks. While still allowing to develop once for more than one platform, Flutter keeps very high performance levels compared to the its biggest competitor React Native, and does not differ significantly from native performances [27].

6.2 Development iterations

In this section, the iterations the app went through are explained. Also follows general information on the process and on the code structure.

Note: the mobile app development took place contemporarily to the physical prototype and website (the backend service, in particular). So, when referring to the sing-in request, signed-in request and BLE communication, details not strictly pertaining the app code will not be expanded on. About the app's code, only relevant details will be explained. The full code is, however, fully accessible on GitHub [58].

Every iteration was executed utilising the Visual Studio Code IDE presented above as the only piece of software necessary for developing Flutter apps for Android. It is important to note that, although the mobile app is cross-platform and thus possible to install on a variety of systems (e.g. both Android and iOS), it was chosen to only test on Android devices. Precisely, it was firstly installed and tested on a OnePlus 5T, then on a OnePlus 9 Pro, both running late versions of the Android operating system.

General code structure

The Flutter app in question has a **main.dart** file which acts as the entry point to start rendering the entire project. From this file, two routes are defined, one for the sign-in logic and one for the dashboard which deals with the interactions with the physical prototype. Main files to manage these routes are **signin.dart** and

dashboard.dart, respectively.

On the single page structure, there is a single object returned which contains all the UI elements the user can interact with in the page. Inside it, many other components are nested to create a UI-tree, that changes its appearance according to the user interaction and external factors - such as fetched data from an API.

6.2.1 Iteration 1. Sign-in flow

During this first iteration, the sign-in pages and handling logic were designed, together with the possibility of keeping the user signed-in when possible. The technology choice for managing these procedures will be explained further, as they are implemented in the RehTracker website project.

In order to be able to contact the backend service, to sign-in and/or verify if already signed-in, the **http** package was added to the project's dependencies [25]. Using this package to perform HTTP requests, two functions are defined in a utility file: **signIn** and **signedIn**. The first one sends username and password to receive an authentication token when the credentials are correct. The latter sends the token, when there is one already saved on the device's secure storage, and returns the result of asking to the backend if the token is still valid, hence validating if the user is still signed-in or if they need to sing-in again. To check if the token is already present from a previous successful sign-in, another utility file defines a series of functions to manage the device's secure storage called **flutter_secure_storage** [75].

Sign-in page

Upon opening the app, first signedIn is called to check for the presence of a previous authentication token. If one is found and the backend declares it to be still valid, the app proceeds to the Dashboard page. In case of a negative reply, the Sign-in page is presented to the user. See figure 6.1 for the flow chart.

From the Sign-in page, if the credentials are correct, the Dashboard is loaded. Otherwise, an error message is shown and no action is taken, as the Dashboard page can only be accessed upon authentication by design choice.

The page's appearance and graphic states are shown in figure 6.2, the code is available in full on the GitHub repository [58].

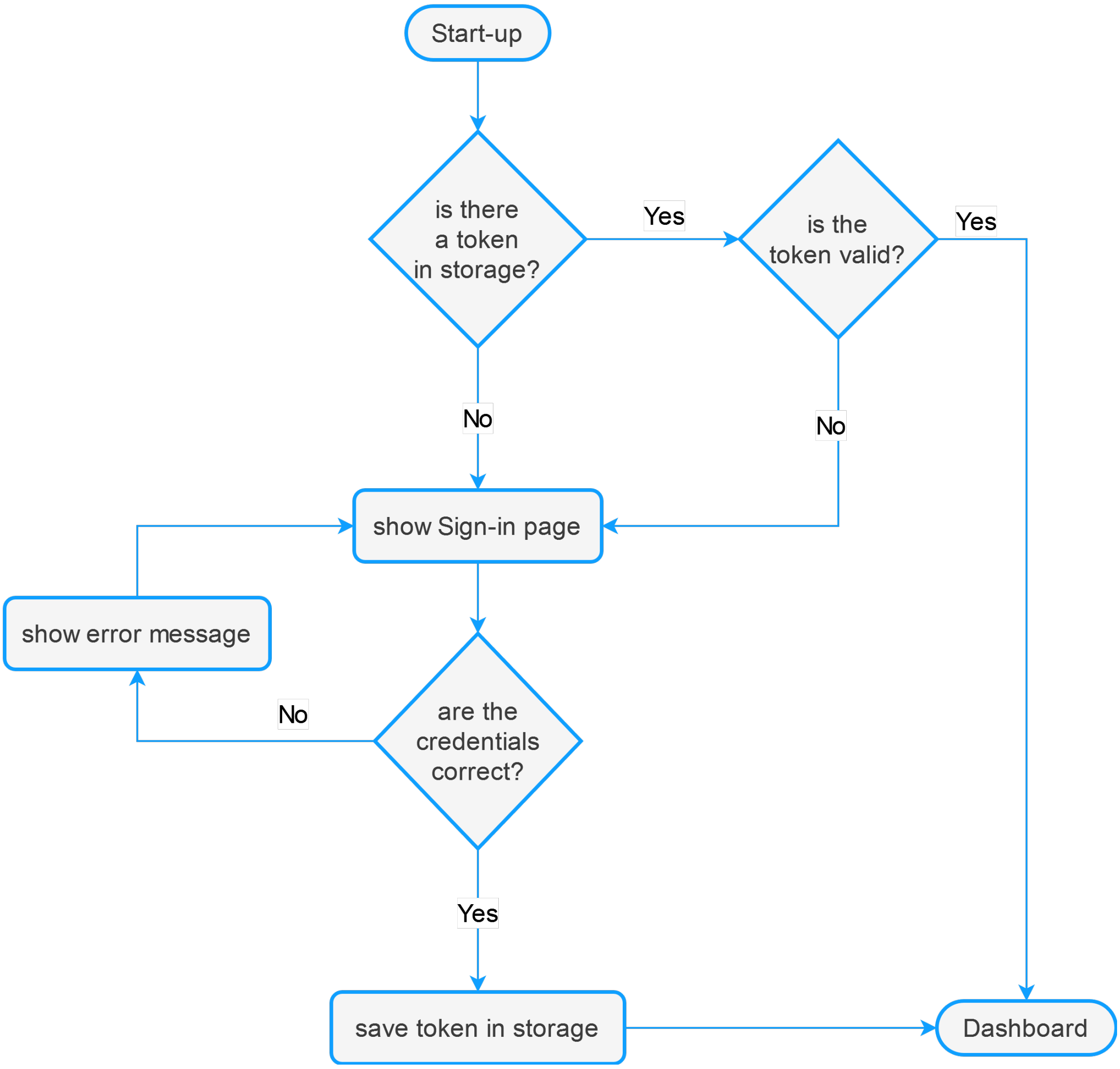
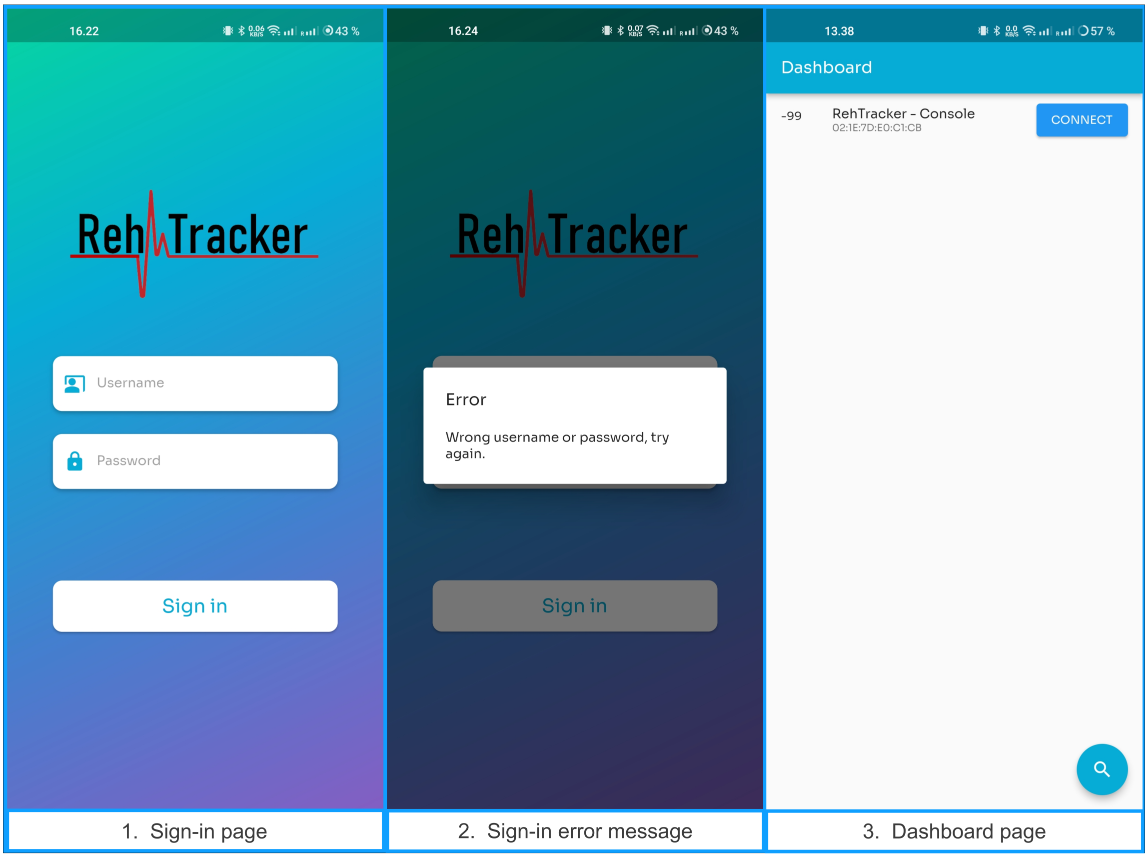


Figure 6.1: The flow chart above shows the signed-in and sign-in flows presented to the user at the app start-up. The Dashboard block represents the opening of the Dashboard page.

6.2.2 Iteration 2. BLE flow

Once established the sign-in flow and pages, it was time to develop the actual interaction with the physical prototype, which was already through the BLE implementation phase.



Error

Wrong username or password, try again.

1. Sign-in page
2. Sign-in error message
3. Dashboard page

Figure 6.2: 1.: a snapshot of the Sign-in page, with input fields for the credentials and a button to trigger the sign-in flow. 2.: a snapshot of the failed sign-in message presented on the Sign-in page. 3.: a snapshot of the Dashboard page accessible after successful sign-in or signed-in requests.
In 1. and 2., an improvised RehTracker logo is visible. The logo is not of any relevance to this document and was designed purely for testing purposes.

To enable the Bluetooth functionality in Flutter, only a small number of packages have been developed. The choice fell on **flutter_blue**, an easy-to-use and rich package for abstracting the concepts of BLE in Dart [26].

Dashboard page

As shown in the sign-in flowchart [fig.: 6.1], consequently to a successful signed-in/sign-in operation, the Dashboard page is presented to the user. Its purpose is to enable the search for BLE peripheral devices and show them in a device tile list, with the possibility to establish a connection to them. Before using any Bluetooth service, the code checks for some necessary Bluetooth and localisation permissions. If those are not granted, the app shows a **BluetoothOffScreen** page, as they are required by Android. Every device tile shows the name of the device and a **Connect** button to start the connection to it.

Then, if the automatic search does not produce any result, a floating button is available to manually start the search process.
Note: only the **RehTracker - Console** device will actually be shown as the app is only intended to work as a part of the RehTracker proof of concept.

After tapping on the Connect button on the device tile in the list, a device-specific page is presented with information on the services and characteristics that the device advertises. Refer to the flow chart in figure 6.3.

DeviceScreen page

Upon opening of the DeviceScreen page, the device's services and characteristics are searched. Meanwhile, the basic device info are shown, such as the device name and the connection state.

When the data about services is received, a list of them is generated. The service advertised by the physical prototype is assigned the name **Exercise**; similarly, its characteristics are mapped as **Value** - a number - and **Type** - a text saying either **Muscle contraction** or **Wrist rotation** according to the exercise selected in the type characteristic. As the user clicks on the **Change** button in line with the Type characteristic, the type characteristic value is written to the BLE peripheral as the complementary value, to change the exercise type and reset the counter. See chapter 5 for the characteristics' definitions [par.: 5.2.1].

Meanwhile, additional utility classes create the correct characteristics instances and take care of subscribing to the exercise value one. This way, the app process automatically updates the exercise counter shown in correspondence of the Value text label. The exercise type value is shown under the Type label, instead, mapped with the text strings previously mentioned, from a true/false value coming via Bluetooth [fig.: 6.4].

Note: the app is only a prototype and a part of a proof of concept. For this reason, the **Start** button is non-functional and only serves as a graphical element to illustrate a future functionality, to start the execution/recording of the exercise [fig.: 6.4].

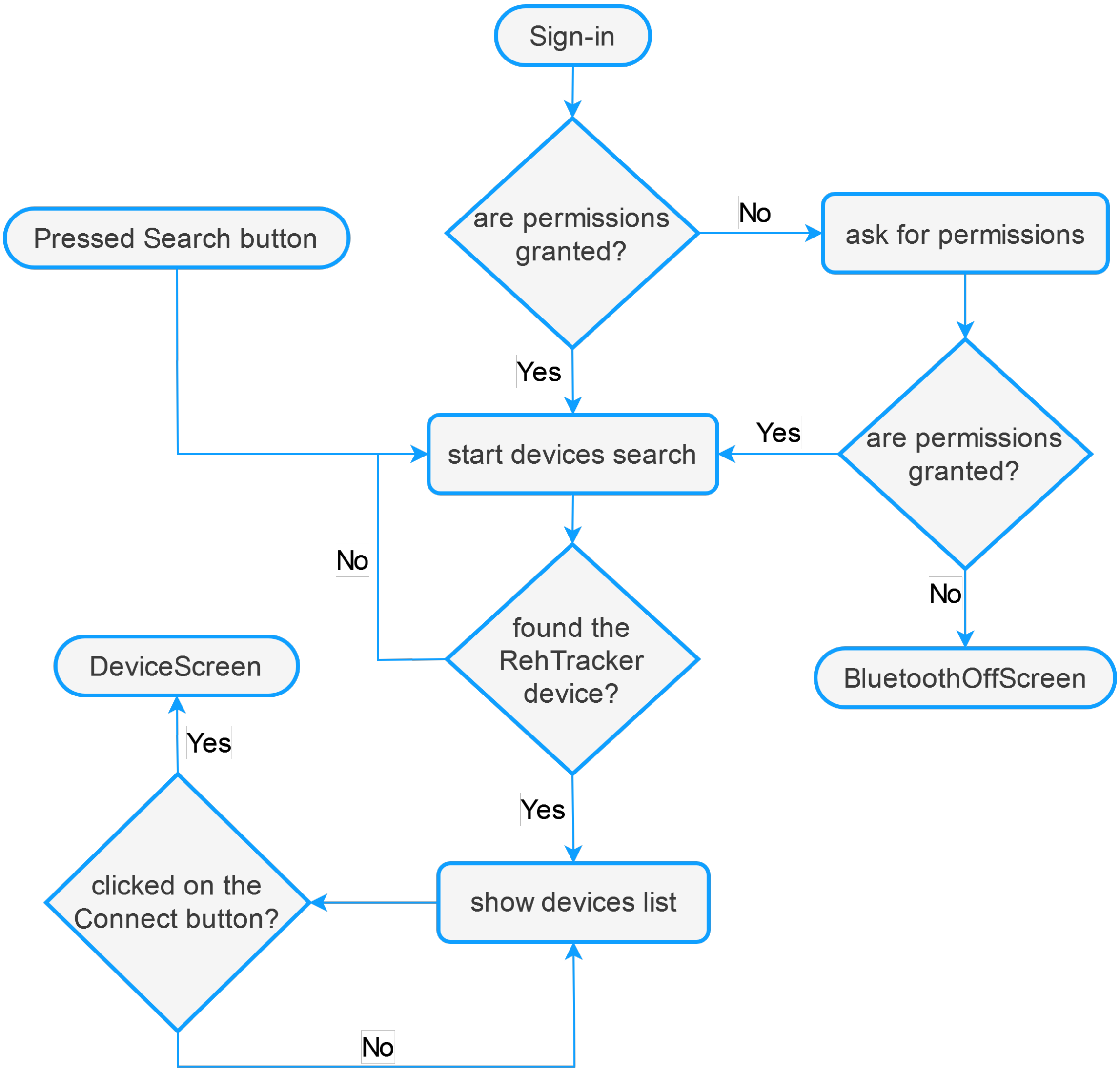
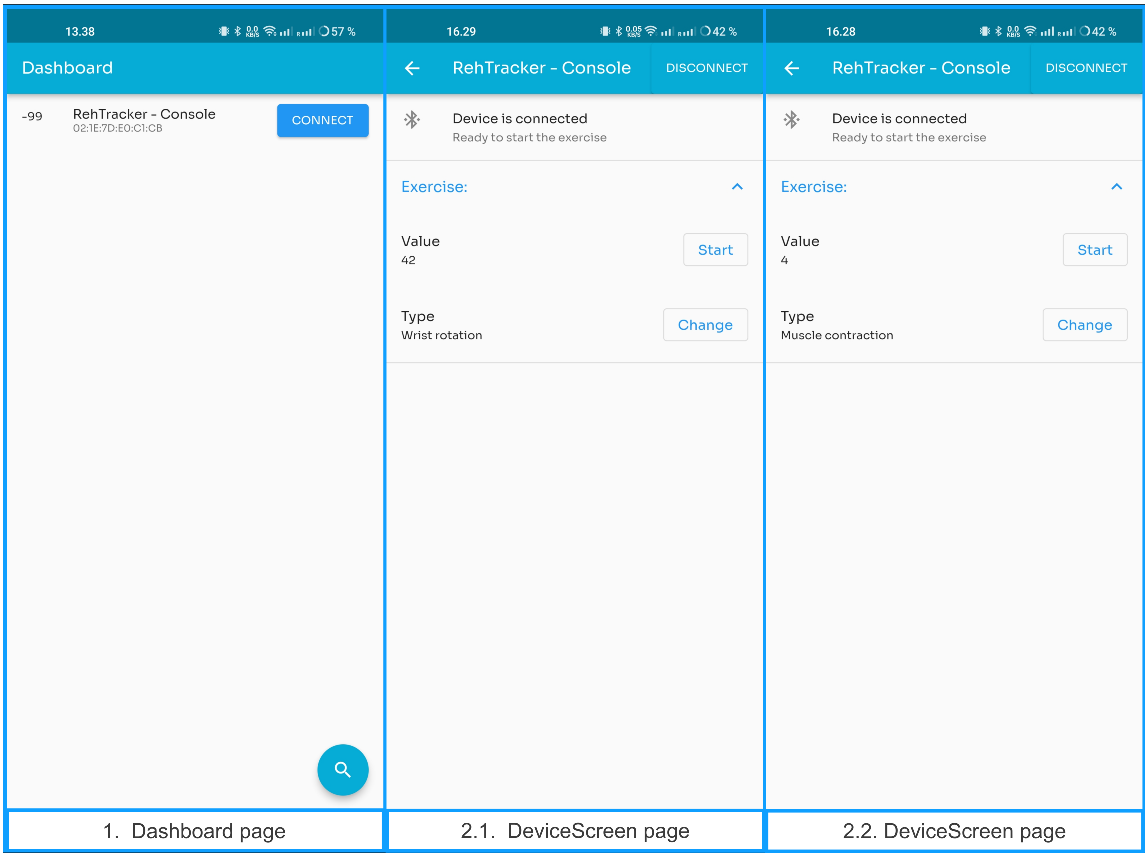


Figure 6.3: The flow chart above shows the possible interactions and flows starting from the opening of the Dashboard page. Pressed Search button refers to the floating search button positioned at the bottom-right corner of the snapshot 3. in figure 6.2. The DeviceScreen block represents the opening of the DeviceScreen page.

BluetoothOffScreen page

To cover the possible state in which the necessary permissions would not be granted to the app, the **BluetoothOffScreen** page was designed. It only servers as a warning page and shows a Bluetooth-off icon paired with a text string indicating the Bluetooth



RehTracker - Console
02:1E:7D:E0:C1:CB

Device is connected
Ready to start the exercise

Value
42

Type
Wrist rotation

Device is connected
Ready to start the exercise

Value
4

Type
Muscle contraction

1. Dashboard page | 2.1. DeviceScreen page | 2.2. DeviceScreen page

Figure 6.4: 1.: a snapshot of the Dashboard page, with the Connect button. 2.1.: a snapshot of the DeviceScreen page with the Wrist rotation type selected. 2.2.: a snapshot of the DeviceScreen page with the Muscle contraction type selected.
Some values are shown in the Value field from a real test of the app-prototype interaction.

state.

This page is also shown whenever the Bluetooth module is turned off - or is in the process of turning on/off - on the device in which the app in running.

6.3 App and physical prototype interaction

Orchestrating the mobile app and the physical prototype's synergy required the app to be completely developed. The physical prototype, however, was ready for testing already at the end of its first iteration [par.: 5.2.1].

As shown in the snapshots [fig.: 6.4], both muscles contraction and wrist rotation exercises were tested using the setup shown in figure 5.9. Specifically, the physical wearable was placed on the left forearm while the app was operated by the right hand.

Testing allowed to spot small correction to apply to the app UI and to the physical prototype's code sketch. After the adjustments, RehTracker was ready to continue with the development of the website.

Note: as a reminder, it is important to underline again that all the three components of RehTracker were developed at the same time. At the beginning of the app development process, for example, the backend service of the website had already been written and deployed to an external hosting service. The advantage of this choice consists in the authentication logic being already available for the app to define its own part of the sign-in/signed-in interactions.

6.4 User testing

As mentioned in 2.1.4, a brief non-expert test was conducted to collect opinions on the interaction between the physical prototype and the mobile app, at the end of their development. For this test, two non-expert students were selected from University of Southern Denmark (SDU).

6.4.1 Structure of the interviews

To start, both the components were explained and the physical prototype was put on the subjects' wrists. Following, similar questions to the late expert reviews were formulated: about the feeling of using the device and seeing the statistics come up on the app, regarding wearing the physical device on their own and, finally, opinions on the visual feedback coming from the LED strip.

6.4.2 Responses

Having the device on the wrist felt light weight and comfortable to wear. However, person two expressed their difficulty understanding the app and pointed to the LEDs' extreme sensitivity.

When they were asked to wear the physical part on their own, both users experienced problems while connecting the pads to the wires or even simply applying them on the skin. Mounting the device on the wrist, instead, was straight-forward.

On the effectiveness of the LEDs, both interviewed agreed that the lights provide a nice way to check on the progress and correctness of the execution. User number two also suggested the implementation of game features in the app.

Outcome

Elaborating on the users' replies, comfort, wearability and lights' effectiveness were confirmed just as it was registered in the late expert reviews. This solidified the premises on which RehTracker has been developed as a potential real-life solution for rehabilitation.

Likewise, the LEDs' excessive sensitivity and the difficulties with the pads appeared again, suggesting ever more strongly the necessity to address these issues in the future.

However, the lights were recognised as effective and suggestions on additional features were given by one of the users. The LEDs were than re-confirmed to be a valuable form of visual feedback, backing up again on the premises for the development of the RehTracker.

Chapter 7

Website development

This chapter describes the independent and mostly simultaneous development phase of the website of RehTracker. The process explained extends from chapter 2 [par.: 2.1.5].

Note: the following sections will often refer to the two main part of the website as **backend** and **frontend**. The former is the user interface app. The latter, instead, contains the API routes and serves the frontend upon user navigation and interaction. The backend also takes care of the communication with the database.

7.1 Technologies

Prior to the actual development of the website and its services, the main technologies were researched and identified. Following are details about this process as well as reasons behind some of these choices.

7.1.1 Node.js, Express and TypeScript

Being this a web application, **Node.js** was identified as the environment of reference to run the code for the website, both backend and frontend [36].

The reasons behind choosing it for this project simply lie in the great diffusion it has, thus meaning that there is a high degree of support from the developer community, and the fact that runs JavaScript code which is easy to program with and very flexible. In fact, it is used by many big companies and is perfectly suitable for small to big-scale projects [61].

Having decided on Node.js to run the website code, **Express** was then picked as a web application framework to easily create a server instance and expose the frontend code from it, while handling the custom API routes [23].

On the matter of the programming language, instead of JavaScript, **TypeScript** was chosen [48]. It allows for a more robust code writing and debugging, while running everywhere plain JavaScript does. The main difference from JavaScript, reason for which it was chosen over the untyped language alternative, is the type support it offers.

7.1.2 MongoDB Atlas and mongoose

For the purpose of creating a completely working example of an authentication flow, and to support the frontend dashboard mock-up, it was decided to utilise a database. The role of such a part in the website would have been taking care of the storage of sign-in credentials and sample exercise information.

Among the number of available database services, **MongoDB Atlas** was picked as a free solution to host a database for testing purposes [51]. It offers a free subscription and a free cluster to host one or more databases, with the possibility to query the database documents from a remote Internet access - e.g. from a backend service running on a server.

The database hosted on MongoDB Atlas are MongoDB collections, JSON-like documents that can be directly represented by the data models in the code of a web application [52]. Hence, this was a better choice than a more traditional relational database - e.g. PostgreSQL and MySQL - as mapping the data collections would have been easier in the website's backend service.

Another advantage of non-relational databases like MongoDB is the higher performance on most data operations, when scaling the web application to a high amount of data to manage. Differently than relational databases, the non-relational ones - also called NoSQL databases - perform operations more efficiently and in a shorter time [37].

Based on the choice of developing for a Node.js environment, it was opted for the package **mongoose** to manage the connection to the remote database and to represent the data types in the codebase [15].

7.1.3 JSON Web Tokens

For the purpose of providing a sign-in and signed-in functionalities to RehTracker's interactions, it was decided to implement a flow based on the exchange of JSON Web Tokens (JWT) [13]. It consists of a security standard that revolves around the idea of a text token, used to authenticate users and subsequently authorise their web requests. The token generated can contain signed information about the user - e.g. the username and the full name - and other details such as the algorithm used to sign it.

In the website, the idea was to generate one of these tokens at a successful sign-in from the user. The token would then be attached to the sign-in response as an **HttpOnly** cookie. This means that it is only readable by the server and not accessible by JavaScript APIs, thus making the frontend website more resistant to attacks [24].

Given the development environment choice discussed previously, the Node.js package **jsonwebtoken** was installed for using JWT on the website [14].

7.1.4 React

For developing the website's frontend, it was decided to create an application with the framework/library **React**. Its purpose is to simplify the frontend development by focusing on the user interface. They way it does that is with the idea of components, units of code which are meant to encapsulate some specific business logic, allowing the composition of more complex UIs based on these smaller and easier-to-code components [42].

7.1.5 Heroku

In order to make the sign-in/signed-in routes available from the app, as well as provide a real-life scenario of a realistic implementation of a solution like RehTracker, it was opted for making the website - more importantly the backend service in it - available and reachable from everywhere.

To achieve such goals, the web application has to be contactable via a public IP address. So, it was decided to deploy the website on a cloud platform. Because of its free-to-try offers and a good integration with GitHub tools, **Heroku** was quickly chosen [43].

Besides providing all the necessary security controls that a web app needs when reachable on the Internet, Heroku makes deploying changes to the app very easy with tools such as automatic deploys when change are detected on a codebase resident on GitHub. Hence, it would have been the perfect service to deploy the website part of RehTracker.

7.2 Development iterations

In this section, the focus will be on the main iterations that brought the final version of the RehTracker website. Full code is available on GitHub [57].

Note: the website development took place at the same time as the physical prototype and mobile app (the backend service, specifically). Thus, when mentioning the mobile sign-in and signed-in routes, refer to chapter 6 for a complete understanding [par.: 6.2.1].

Every iteration was carried out using the Visual Studio Code IDE presented in chapter 6 [par.: 6.1.1].

General code structure

The code is divided up into two **packages**. The package **Frontend** contains and runs the frontend part of the website, the React app. The backend service is contained in the package **Server**, instead. Note: the latter is also referred to as a service, that is because it is always running and waiting for HTTP requests.

In both packages, a **src** folders contains the files that define the business logic. This folder will also be called **source folder**. Additional files are placed aside this folder, containing information on the packages' dependencies and useful commands.

Inside the Server's source folder, a subfolder called **models** hosts the model definitions for the MongoDB database.

7.2.1 Iteration 1. Server

To start the development process, also to support the development of the mobile app, the Server package was created. One central code script called **server.ts** acts as the starting point of the service.

API request path	HTTP request type	Device source
/sign-in	POST	web
/mobile/sign-in	POST	mobile
/signed-in	GET	web
/mobile/signed-in	POST	mobile
/sign-out	GET	web

Table 7.1: The main API routes defined for managing the authentication and authorisation flows on the backend.

Initial definitions

This script first creates an Express app, then defines some properties such as the default JSON format of the responses' bodies and the cookie policies, which will be needed when defining the sign-in and signed-in routes.

Following, the code for connecting to the database is set-up. The actual connection will, however, only be possible after defining the database with MongoDB Atlas.

Finally, the server process is set to listening mode on a specific port. Before these lines, the frontend will be served when developed.

Routes

In this paragraph, the most relevant routes will be explained. More routes are defined in the project, yet they will not be covered in this document because they are not in use in the scope of the proof of concept.

Using the **Router** module from the Express package, a router object is initialised. It can handle many HTTP request types, with the possibility to use a middleware for pre-validation or security checks prior to the request handling, e.g. an authorisation check.

Some API routes are defined afterwards, of relevance is the **AUTH_PREFIX** router prefix which will identify every endpoint of the authorisation/authentication type. That is to say that every request of those kinds will have to be prepended by the text string **/api/auth**.

Table 7.1 shows the main routes and their target source, **mobile** (from the app) or **web** (from the frontend).

Sign-in routes: these routes, both web and mobile versions, use the database model's method to check the password contained in the request, by comparing it with the encrypted version stored in the database. In case of a successful comparison, a JWT token is generated and sent back with the HTTP response, using the jsonwebtoken package. The token is created with an embedded duration time - 1 hour for developing and 10 minutes for the published website - after which it would expire, thus failing the verification.

For the mobile route, instead of sending the token in an HttpOnly cookie, it is sent a the response body. This was due to complications in implementing access to cookies from Android, using Flutter.

The passwords are signed in the creation phase using an environment secret that is not visible to the outside of the machine that is running the Server code. The same secret is then used in these API routes to verify the password with its signed version.

Signed-in routes: here, for the web route, the request is checked to see if there still is a token cookie. If it is there, a check is run with the **verify** method of jsonwebtoken and the user's username is returned to the frontend in case of success. Otherwise, an empty username is sent back - i.e. a null instance in TypeScript.

In the code for the mobile route, the logic to check the token is identical to the web's route. However, the token comes from the mobile app as part of the body of the request and the reply is simply a boolean value: true if the token is still valid to prove the user's authenticity, false if the token is not present or expired.

Sign-out route: to enrich the frontend UI, it was decided to add a sign-out functionality. This route just removes the authentication token from the cookies, causing future signed-in request to return a negative reply.

Other routes: a handful of additional routes has been added to provide the options to fetch exercise information or create user accounts. Although most of these additional routes are not used in the proof of concept of RehTracker, one specific route turned out to be useful to create a more realistic mock-up frontend dashboard: the **/api/exercises/get** route will be used to fetch the exercise statistics per date, given a username and verified the authorisation token.

Authentication middleware

With the purpose of providing a more complete example of a web application's secure implementation, an authentication middleware was created.

It consists of a function simply named **auth** which verifies the presence and validity of a JWT token cookie. Then, the actual request handler takes control of the request, in case of a positive check. In the negative scenario, the auth middleware replies automatically with an error.

This middleware will be used when handling the /api/exercises/get route coming from the frontend.

7.2.2 Iteration 2. Database

On the grounds of the technological choices mentioned earlier in this chapter, it was time to create a basic database and lay down the data models for RehTracker.

Models definition

Before implementing the database data structures and the corresponding data types in the backend code, it was necessary to define which data classes were needed for the project.

So, first of all, the class **Patient** was designed. Necessary information to be contained in this class are **username**, **password** - an encrypted version and not its plain text - and a method to compare the signed password when attempting to sign-in, this method was defined as **comparePassword**. Some additional fields were added for completeness, although they don't play any role in this first version of RehTracker, so they will not be discussed [fig.: 7.1].

The **Exercise** class was next, containing a username field to link to the Patient that is executing or executed it and a handful of additional properties [fig.: 7.1]:

- ○ **timestamp:** a date field, the creation date of the exercise instance.
- ○ **type:** a number, the type of the exercise, between a few hard-coded exercise types with the only purpose of testing the frontend later on.
- ○ **currentSet:** a number, represents the current set the Patient is working on.

- **repetitionInSet:** a number, stores the current number of complete repetitions in the ongoing set.
- **therapyId:** will be used to reference to the next model class for fetching statistical data in the frontend.

In order to organise the exercises in a time-managed fashion, an additional model was introduced: **TherapyPhase**. The idea behind it is associating the fixed exercise types - again, the ones defined purely for the frontend mock-up - to a Patient and to a specific time interval - i.e. from start to end of a therapy sessions. Doing so, it would be possible to assign different kinds of exercises to a patient with respect to different phases of the rehabilitation, here called TherapyPhases [fig.: 7.1].

As mentioned in 7.1.2, the code implementation follows almost exactly the database models. This way, from the Server, it was possible to implement the model definitions right away, in the source folder's subfolder called **models** [57].

Database creation on MongoDB Atlas

Finally, the database was created on the MongoDB Atlas web portal. Then, it was only necessary to change the database settings to make it accessible from the Internet - i.e. from the backend - through a shared password access. This password is saved on the Server as an environment secret. A similar procedure is explained on MongoDB's documentation online [50].

7.2.3 Iteration 3. Frontend mock-up

The last iteration focused on realising a frontend application to create a visualisation of the exercises that a patient is performing/has completed.

AuthContextProvider

In order to implement the frontend-side authentication logic, a React context is defined with the name **AuthContextProvider**. The entire app is then wrapped in this context component which hosts some key states and functions [57]:

- **authStatus & getAuthStatus:** a variable containing the user's username or nothing;
  a function that updates authStatus by calling the API at /api/auth/signed-in.

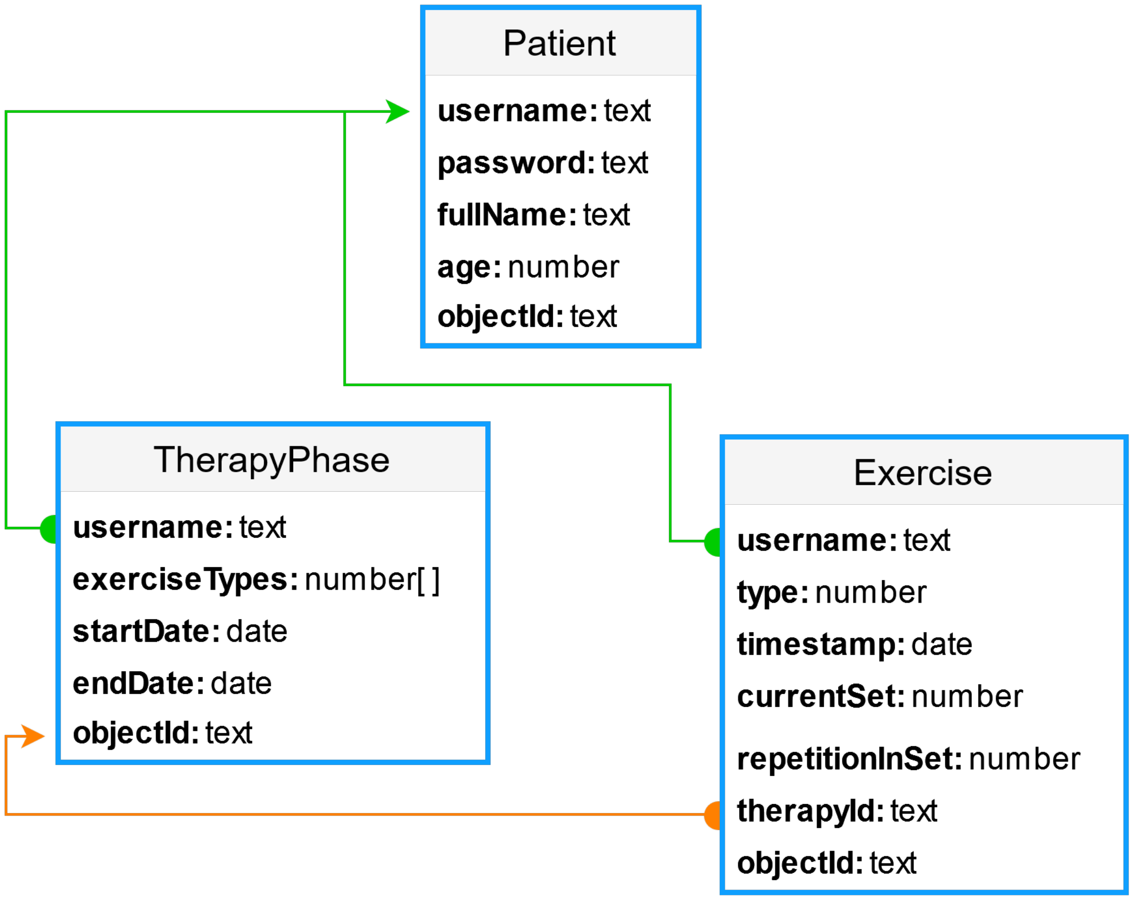


Figure 7.1: A scheme of the database models, each model is automatically assigned an **objectId** which identifies single instances uniquely. However, the Patient's username is also created as unique per Patient, from the backend. The arrow show the referencing relations between the models.

- **globalDate & setGlobalDate:** a variable carrying the date set for fetching the exercise info - defaults to the current date;
  a function used across the app to set the date.
- **stats & getStats:** an array containing the exercises' information fetched through the exercises API [par.: 7.2.1];

Router

Using the package **react-router-dom**, a general Router is defined in the **components** folder. This router element allows to define which React components - i.e. pages - to render on the frontend, according to the route the user is visiting [73].

The initial route, as well as every unknown route, goes to the homepage: a **Home** component which only triggers the re-fetch of the authStatus. If this status contains the username, meaning that the user is signed-in, a **/stats** and **/sign-out** routes are enabled. The former leads to a **Statistics** component, the latter triggers the sign-out flow.

In case the authStatus does not contain a username, the **/sign-in** route becomes active and a sign-in button is rendered in the **Navbar**. The Navbar is a simple navigation bar at the top of the page with links to the routes described earlier and a button spot, either for signing in or signing out [fig.: 7.2].

Figure 7.2: Top: a snapshot of the Navbar component before the user's authentication. Bottom: a snapshot of the Navbar component after successful user sign-in; the Statistics page becomes accessible and its link visible.

Sign-in page

After clicking on the sign-in button, the user is presented a simple form with fields for username and password. Upon clicking the sign-in button in the form, the user is presented with an error message or redirected to the Statistics page, accordingly with the sign-in failure or success.

In case of successful authentication, the authStatus is also updated.

Statistics page

The most important page of the frontend is the Statistics page. It shows a calendar component that allows the user to select a specific date, after which the globalDate is updated.

When this happens, through a call to the /api/exercises/get route, the stats property of the AuthContextProvider is populated with the response from the backend. If there are TherapyPhase and Exercise instances for that user in the specified date, they are fetched in an array-like structure.

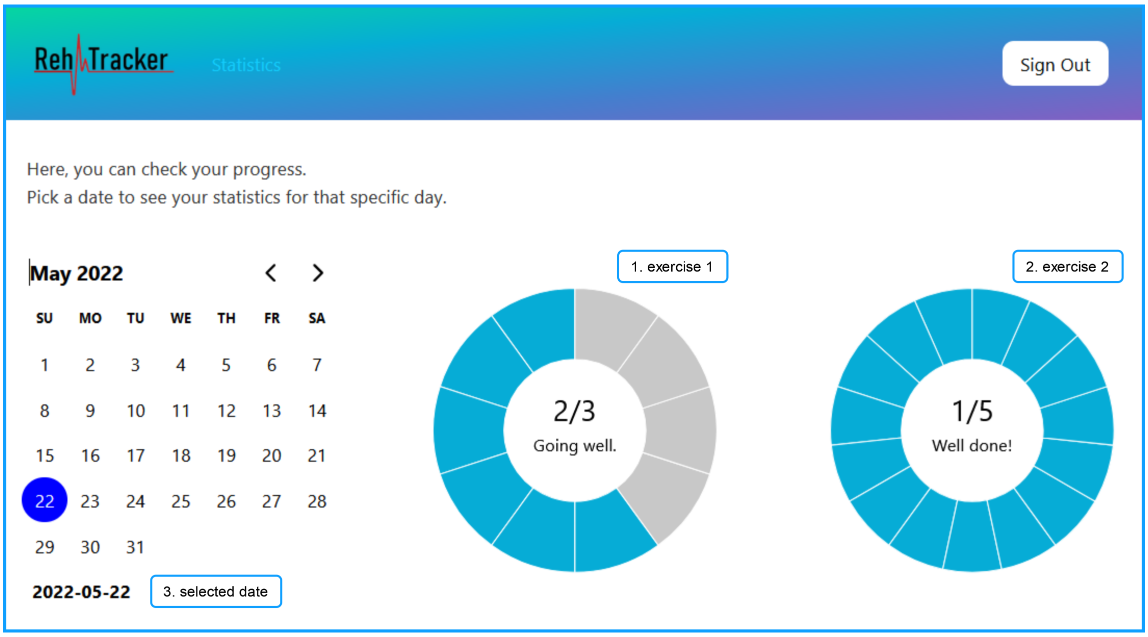
After receiving the exercises, for each of them, a pie chart is shown to represent the progress level on that specific exercise. The charts are made using the Node package **recharts**, which allows for a high customisability [65].

The charts show the completion status of the current set on the outer coloured section. On the inside, a fraction displays the number of completed sets of repetitions over the expected total [fig.: 7.3].

The stats are cyclically re-fetched using the shared method getStats every sixty seconds, to ensure the latest exercise statistics are shown on the page.

7.3 RehTracker overview

As the website frontend was completed, together with the mobile application and the physical prototype, the initial idea behind RehTracker was accomplished. A complete ecosystem was developed to serve as a proof of concept of a promising novel approach to foster motivation in patient undergoing rehabilitation of the forearm. A diagram of the complete interaction is available in figure 7.4.



Here, you can check your progress.
Pick a date to see your statistics for that specific day.

SU	MO	TU	WE	TH	FR	SA
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

2022-05-22

Figure 7.3: A snapshot of a test user and their statistics when picking a date. 1.: First exercise performed on the selected date. 2.: Second exercise performed on the selected date. 3.: Selected date from the date picker, printed out at the bottom of the calendar view.

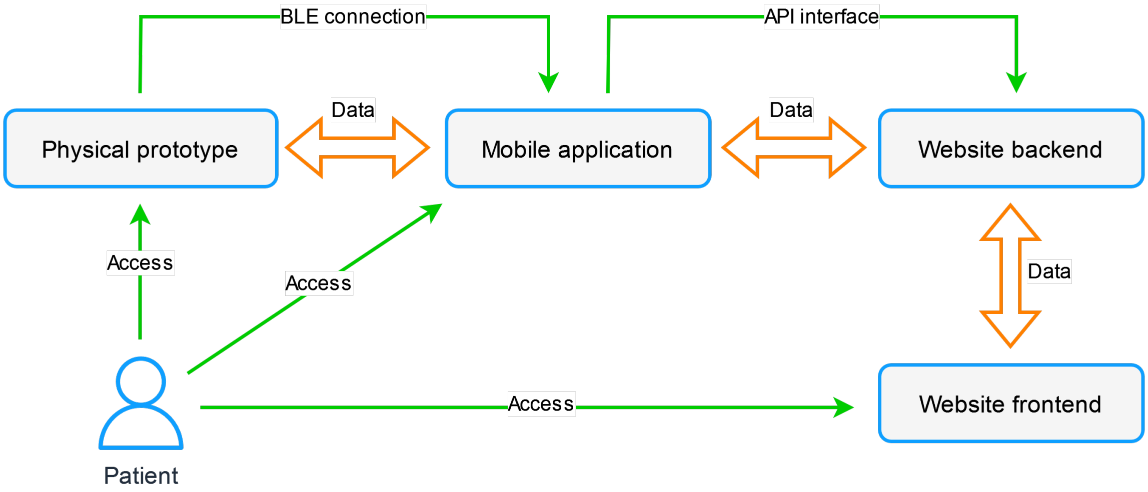


Figure 7.4: A diagram of the entire ecosystem and the ways its component interact together. The main actor is the patient.

Chapter 8

Robotic visualisation development

This chapter explains the development of RehTracker's robotic visualisation mode. The process explained continues from its basic structure in chapter 2 [par.: 2.1.6].

Note: the following sections will point multiple times to the physical product. For a complete understanding, refer to chapter 5 [par.: 5].

8.1 Technologies

The development started from an already-developed **Simulink** model featuring a 3D visualisation of the UB Hand IV and the necessary Simulink blocks to calculate an accurate closing animation for all the hand's joints [47].

8.1.1 MATLAB and Simulink

The main model was adapted starting from the initially developed design to control the UB Hand IV. Thus, Simulink was used to re-define the model for this visualisation. In general, this tool offers an environment to design and simulate hardware-software interactions using block diagrams [79].

For communicating with the physical prototype, it was necessary to write a code script using **MATLAB** as it allows to inject data into the Simulink model while it is running [78].

ble interface

Reading data from the MATLAB script through BLE was possible thanks to the MATLAB **ble** interface. It defines a handful of functions to interact with a BLE device and map its entities - e.g. the **characteristic** function allows to define an object for reading/writing data [77].

8.2 Development phases

Following, the two phases that led to a working solution for the robotics visualisation are described. The full code and Simulink model files are available on GitHub [59].

Note: the whole development was performed on a PC running Windows 10 equipped with a Bluetooth interface, running the MATLAB code locally on the same machine.

8.2.1 Phase 1. BLE communication

In this first step, the necessary MATLAB code was implemented to enable the BLE connection to the physical prototype. Concurrently, the Arduino code was updated to support this new operating mode.

MATLAB and BLE

In the main code file called **ble_utils.m**, the main logic was implemented by defining a **device** interface - using the physical prototype device ID - as well as the two characteristics needed: **ch_data** and **ch_robot**.

The purpose of ch_data is to contain the data on the muscle contraction level, whenever a new value is notified from the BLE peripheral device. The characteristic ch_robot is, instead, used to write to the Arduino that the robotics visualisation mode is selected, instead of the default RehTracker exercises.

After the initial definitions, a value of **1** is written to the ch_robot characteristic. Finally, a custom handler for new ch_data values is assigned to the characteristic property responsible for handling the incoming data stream.

A separate code file called **readBleDataCallback.m** contains the handler definition referred to in the main file as the **readBleDataCallback** function. Here, the newest available value in the BLE stream is read, converted into a number from 0 to 1 and assigned to an environment variable called **data**, accessible from the Simulink model.

Arduino mode

Note: in the Arduino code, the robotics visualisation mode is named robotics demo. So, following, this name will be used to refer to it.

The Arduino sketch was updated to support a new BLECharacteristic that contains a boolean value: if true (set to 1) the **robotics demo** is selected, otherwise (set to 0) the default behaviour applies. The setup code follows the same structure as for the already-defined characteristics.

In the loop section of the code, before the lines that manage the two exercise described in chapter 5 [par.: 5], a new check is added to read from the robotics demo characteristics and activate the code to handle it.

The handling code is isolated in the **handleRoboticsDemo** function which simply reads the contraction level from the myosensor and updates the related characteristic value with a percentage of contraction, calculated over the maximum level defined in 5.2.3.

MATLAB and Arduino interaction

Upon testing both the board and the MATLAB script, potential mistakes were corrected and optimisations were applied. Therefore, it was time to work on the Simulink model to complete the data's journey: from the muscle sensor, trough the BLE communication channel, to the MATLAB runtime environment and finally to the model's virtual hand.

8.2.2 Phase 2. Real-time data visualisation

Following phase 1., this second step dealt with the challenges of adapting the Simulink diagram to digest the real-time data coming from the code script.

Model structure

The model is represented by a block diagram, divided up into three main areas:

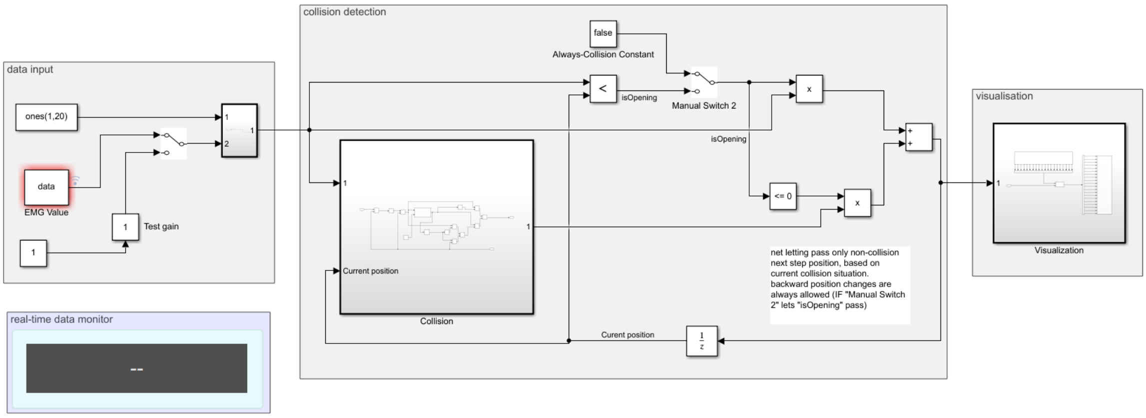
- **data input:** feeds the EMG input coming via Bluetooth to the next area.
- **collision detection:** calculates if the hand 3D model would be colliding with the new input position, if not the value goes on to the next area.
- **visualisation:** converts the input data into VR coordinates for the hand model.

Simulation flow

At the beginning of the diagram, the data input area contains a constant block connected to the data environment variable defined in MATLAB, which will be hosting the current contraction value as it comes real-time from the Arduino. The value is multiplied by a one-dimensional array of twenty ones - **ones(1,20)**) in the code - to spread the contraction level across all the fingers [fig.: 8.1]. This is possible by taking advantage of a single hand synergy for grasp control as Meattini et al. [46] have successfully been able to do in building an sEMG-based HRI for robotic hands, using synergies and machine learning.

Following the input section, the collision detection area takes care of detecting collisions both between the hand fingers and a 3D rectangular prism and finger-to-finger, to stop the closing animation [fig.: 8.1].

Finally, the visualisation area transforms the synergy coordinates into VR coordinates to feed to a VR Sink which then animates the 3D hand model accordingly [fig.: 8.1].



net letting pass only non-collision next step position, based on current collision situation. backward position changes are always allowed (IF "Manual Switch 2" lets "isOpening" pass)

Figure 8.1: The Simulink diagram shows the three processing areas in grey. A constant called **Always-Collision Constant** forces the collision detection network to always check for collisions, thus making the opening animation smooth as the closing one. A real-time data monitor shows the EMG value while it gets updated, for testing purposes.

The hand 3D model closes by wrapping around the orange bar, as shown in figure 8.2.

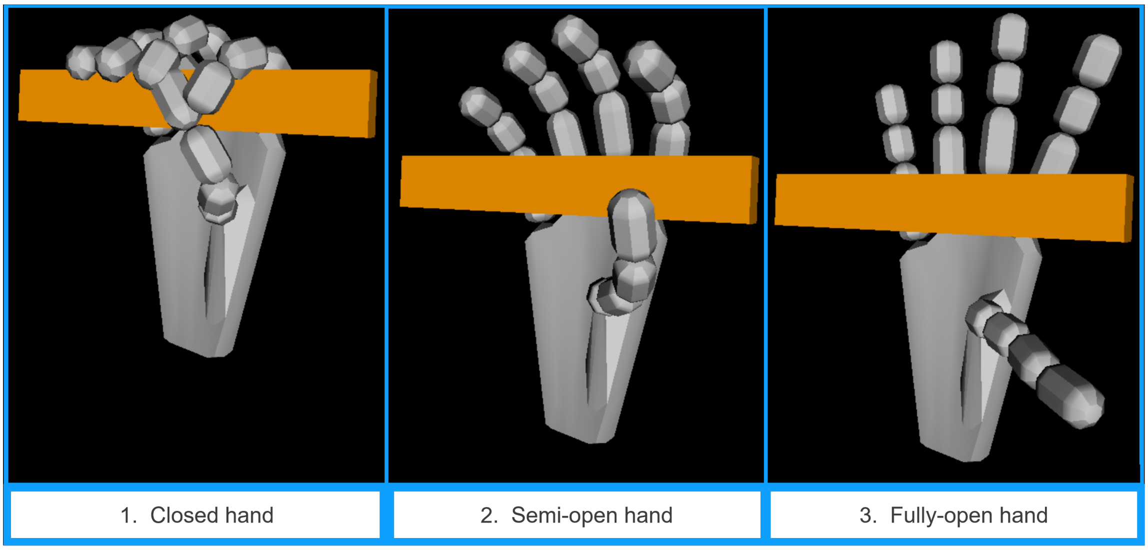


Figure 8.2: Snapshots of the visualisation step of the Simulink model while it is running. 1.: the contraction is maximum and the hand is completely closed. 2.: the contraction is at medium intensity and the hand is semi-open. 3.: the contraction is absent and the hand is fully-open.

8.2.3 Final testing

To end this development process, a thorough technical test of the MATLAB-Arduino interaction followed. Modifications and optimisation were applied were necessary. No further testing was necessary as the robotics visualisation is designed to only interact with the physical prototype.

The end of phase 2., comprehensive of the testing phase here mentioned, marked the end of the development of the RehTracker, inclusive of the robotics visualisation mode.

Chapter 9

Results

This project produced two results: the development of RehTracker as proof of concept of a rehabilitation aid ecosystem; the robotics visualisation as an example of integration of the UB Hand IV model with the physical prototype real-time monitoring to provide a realistic visual feedback to the patient.

9.1 The ecosystem

At its current state, the ecosystem is constituted of a physical prototype for data measurement and real-time feedback, a mobile app for data reading and longer-term feedback in the form of the exercise's statistics, a mock-up version of a website to include long-term feedback and statistics to the patients.

9.1.1 The physical prototype

The physical prototype is easily extendable because of its contained size and the use of a flexible board, which allows more sensors and code to be added in the future or a reconfiguration of the installed ones. Its features have been subjected to expert reviews and user testing and some weaknesses, possible improvements, have been identified. However, a real-case-scenario test with real patients is still missing. Such tests and interviews would allow for a more precise targeting of the prototype's flaws, to point towards a better solution.

9.1.2 The mobile app

The mobile app can also be extended to support more exercises and statistics, as it has been development using a framework and language based on components. Nevertheless, it is clear that its usability can benefit from improvements as suggested during the user testing phase.

9.1.3 The website

Once again, this part of the project is easy to expand as it has been developed using modern component-based approaches, both for the frontend and the backend parts. Tests have not been run on the website as it only serves the purpose of supporting the app backend functionalities - i.e. the use of the APIs - and of providing a more complete outlook to the global proof of concept.

9.2 The robotics visualisation

To include more visual feedback, the robotics visualisation integrates with the physical prototype through an ad-hoc working mode and represents real-time muscle contraction data on a 3D hand model.

The visualisation explores on the concept of exergames which is not implemented in RehTracker, at the moment. Nonetheless, the virtual environment, constituted of the 3D hand model, provides the basis of a possible exergame scenario to explore in future versions of the project.

As part of the proof of concept, the robotics visualisation has not been subjected to testing or review, to this moment. However, it will be possible to do so once an exergame implementation will have been realised.

9.3 Full workflow simulation

For the purpose of providing a full example of the RehTracker workflow, from an exercise prescription to the monitoring and visualisation, the author includes a simulation of a realistic use case. The left forearm was targeted.

The exercises

Two basic exercises were designed in accordance to the device's placement on the wrist as shown in figures 5.9, 5.12. Exercise one involves the hand's closing con-

Exercise	Exercise type	N. sets	N. repetitions	Done/Total
1	Muscle contraction	1	3	3/3
2	Wrist rotation	1	5	3/5

Table 9.1: Details on the test exercises designed for the full workflow simulation. The exercise type refers to the types' definitions from chapter 5 [par.: 5.2.1].

traction followed by a relaxation/hand opening. The second one is a wrist rotation exercise from one side to the other. Sets and repetitions were designed as shown in table 9.1.

In order to show a realistic setup, the exercises were not both executed to completion. Exercise 2 was executed up to repetitions three out of the total of five (see last column of table 9.1).

The signals processed in the physical prototype

For the muscle contraction exercise, the Serial Plotter tool of the Arduino IDE was used to show the recorded contraction signals for two of the three repetitions performed. The graph in figure 9.1 shows two consecutive contractions and the preprocessed signal coming from the EMG sensor, together with some reference values.

As it is visible in the graph, the myosensor tracks constantly the contraction level. Then, the code evaluates if a strong-enough contraction has been made and, at the end of the contraction, sends the updated contraction counter to the mobile app.

As for the rotation exercise, the same method for capturing the data was applied. The graph in figure 9.2 shows the data coming from the IMU as opposite direction rotations on the forearm axis, given the zero y-value as the starting position and the neutral rotation reference. Two of the three performed rotations are graphed, as well as the relevant thresholds and reference values.

The data received in the app

While recording the data, the Arduino was sending the updated counters to the mobile app by updating the related BLE characteristics, as discussed in 5.2.1, see figure 9.3. After completing three repetitions for Exercise 1, the button Change was pressed and the counter was automatically reset to 0. Then, three rotations were performed while the counter updated up to 3.

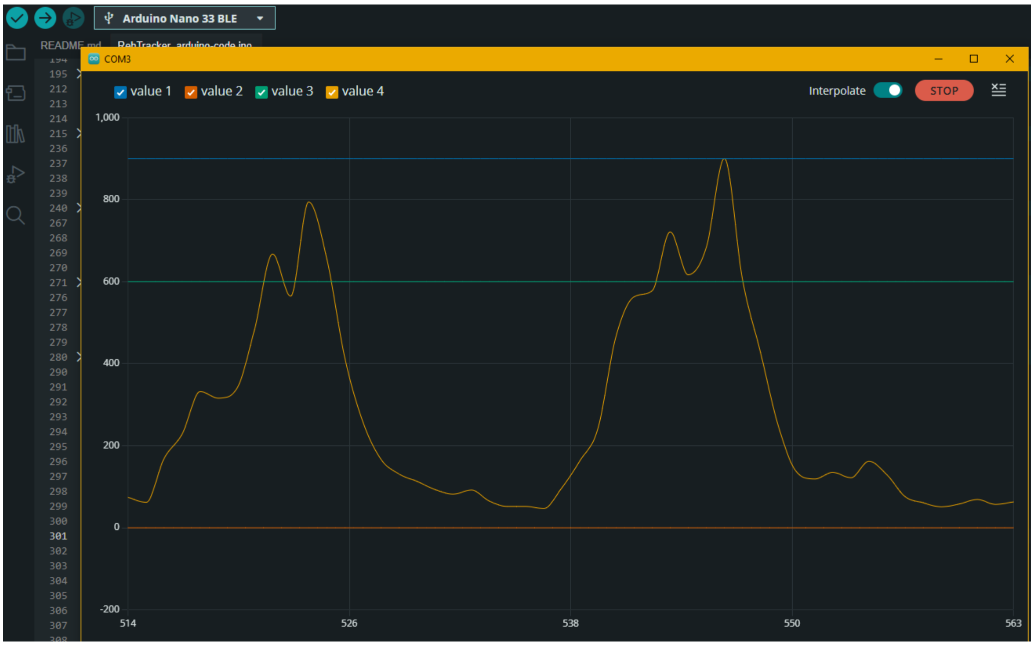


Figure 9.1: Serial plot of the EMG data. In blue: the maximum contraction level set in the code. In orange: the zero contraction level reference. In green: the threshold set in the code to count the contraction as executed. In yellow: the actual contraction data points interpolated using the built-in interpolation function of the plotter.

The statistics in the website and database

As mentioned in chapter 7, the dashboard available on the website is just a mock-up of the potentially working final product. For this reason, the exercises' data on the database were manually inserted using the MongoDB Atlas online tool [fig.: 9.4].

To associate the exercises to the test patient, a TherapyPhase was also manually added to the correct table on the database [fig.: 9.5].

On the website, to simulate the user's behaviour intended for this project, the author signed in the website and the statistics page was brought up, showing the exercises 1 and 2 and their completion graphs [fig.: 9.6].

The robotics visualisation

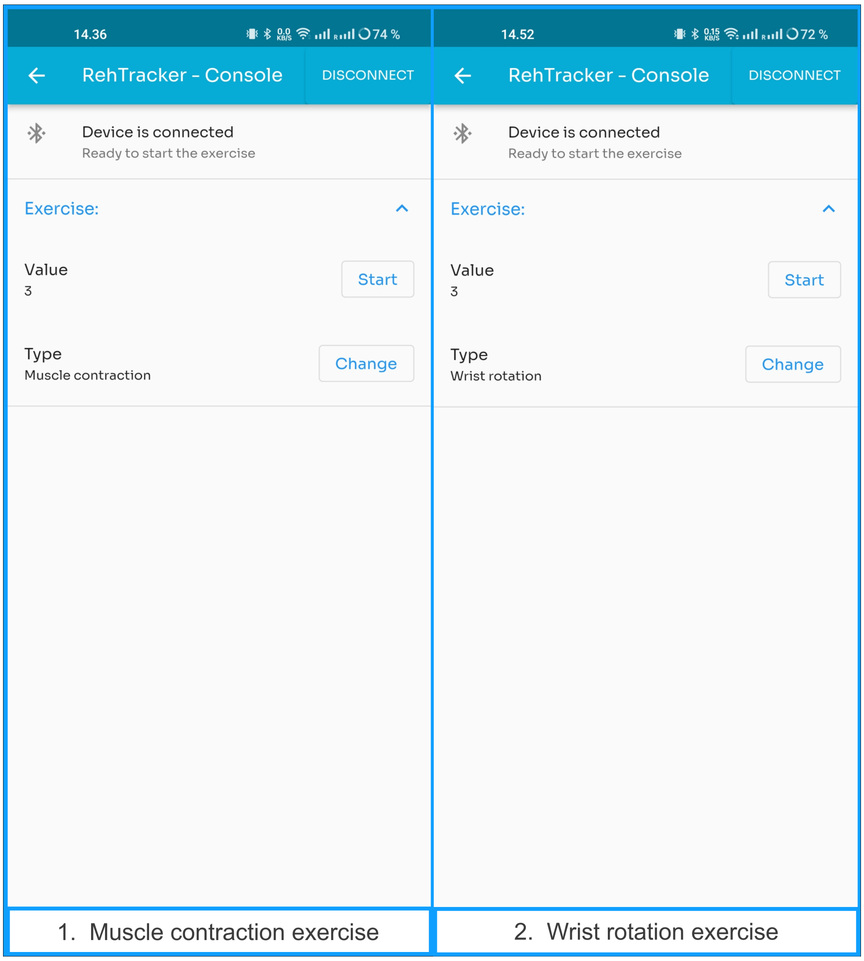
At the current stage of development, it is not possible to visualise the robotic hand model while using RehTracker in the default mode. However, in order to show the

Figure 9.2: Serial plot of the IMU data. In blue and orange: the maximum rotation levels in both directions. In green and yellow: the minimum rotation thresholds in both directions, after which the rotation is counted as completed. In purple: the actual rotation data points interpolated using the built-in interpolation function of the plotter.

data that is fed into the Simulink model in a realistic muscle contraction repetition scenario, the author has extrapolated the EMG data from the first contraction at figure 9.1 and fed the corresponding data conversions to the interval [0,1] to the model.

Figure 9.7 shows the data points during the contraction, sampled at regular intervals 20%-long, and the corresponding hand configurations that the robotic visualisation would produce.

To make the robotics visualisation workflow more clear, a video demonstration has been recorded and is publicly available on YouTube [60].



Device is connected
Ready to start the exercise

Value
3

Type
Muscle contraction

Device is connected
Ready to start the exercise

Value
3

Type
Wrist rotation

1. Muscle contraction exercise

2. Wrist rotation exercise

Figure 9.3: Snapshots from the mobile app while executing the exercises. 1.: the app view during Exercise 1. 2.: the app view during Exercise 2.

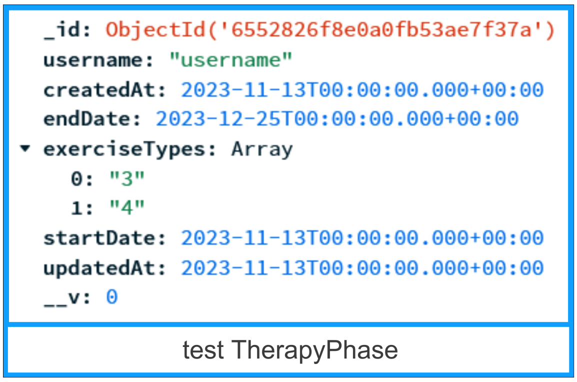
```
_id: ObjectId('655284c48e0a0fb53ae7f37b')
username: "username"
timestamp: 2023-11-14T11:51:00.751+00:00
type: 3
currentSet: 1
repetitionInSet: 3
therapyId: "6552826f8e0a0fb53ae7f37a"
createdAt: 2023-11-14T11:51:00.751+00:00
updatedAt: 2023-11-14T11:51:00.751+00:00
__v: 0
```

1. Muscle contraction exercise

```
_id: ObjectId('655284f58e0a0fb53ae7f37c')
username: "username"
timestamp: 2023-11-14T11:51:00.751+00:00
type: 4
currentSet: 1
repetitionInSet: 3
therapyId: "6552826f8e0a0fb53ae7f37a"
createdAt: 2023-11-14T11:51:00.751+00:00
updatedAt: 2023-11-14T11:51:00.751+00:00
__v: 0
```

2. Wrist rotation exercise

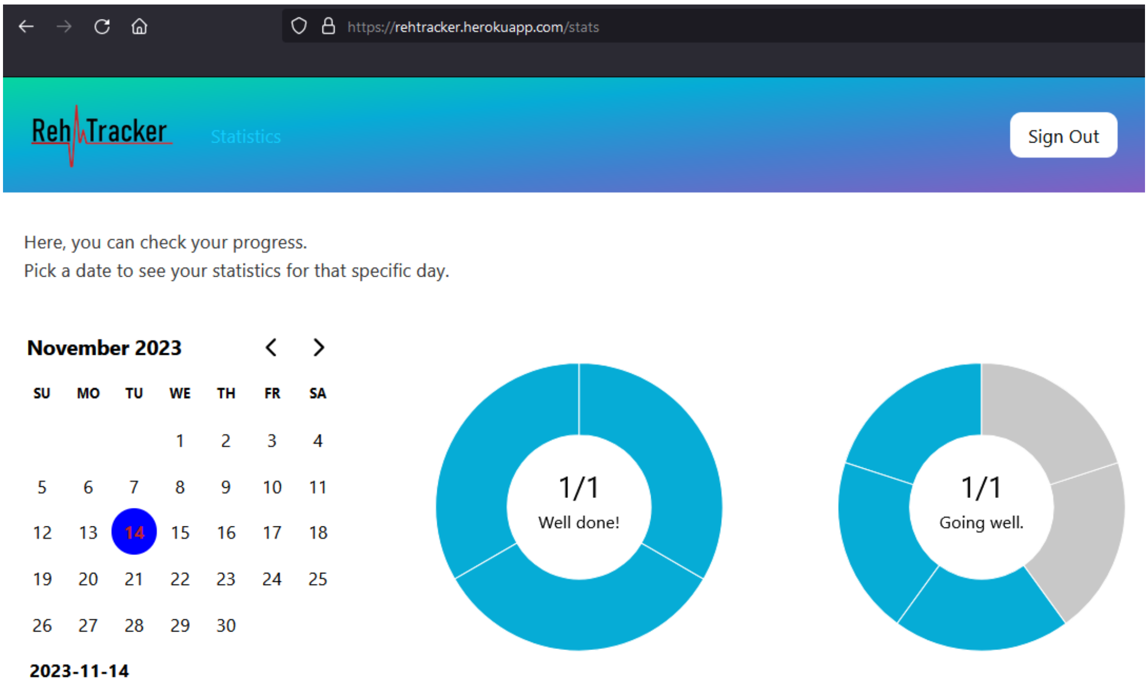
Figure 9.4: The exercises' object representations on the database. The type (3 or 4) is only necessary for the frontend to distinguish these new exercise types from the previously-defined test exercises [par.: 7.2.2]. The therapyId property points to the TherapyPhase connected to these exercises.



```
_id: ObjectId('6552826f8e0a0fb53ae7f37a')
username: "username"
createdAt: 2023-11-13T00:00:00.000+00:00
endDate: 2023-12-25T00:00:00.000+00:00
▾ exerciseTypes: Array
    0: "3"
    1: "4"
startDate: 2023-11-13T00:00:00.000+00:00
updatedAt: 2023-11-13T00:00:00.000+00:00
__v: 0
```

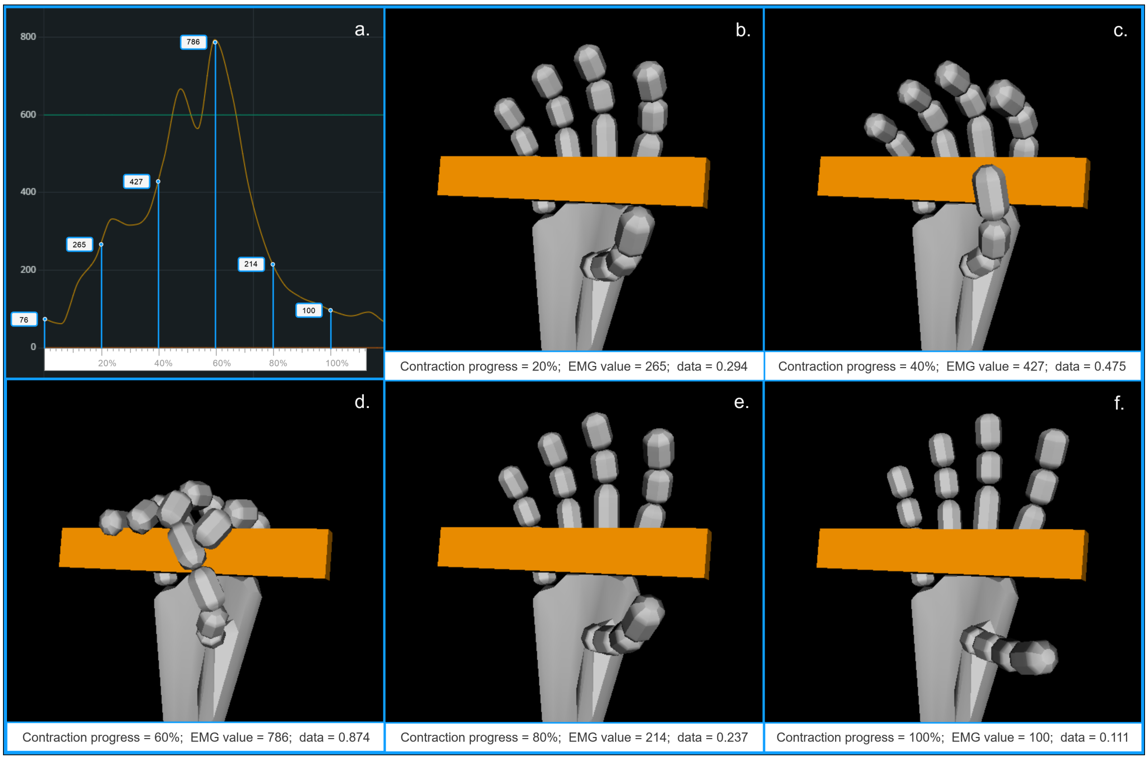
test TherapyPhase

Figure 9.5: The therapy phase's object representations on the database. This instance associates together the Patient with username "username", the exercises of types "3" and "4" and the therapy period identified by "startDate" and "endDate" - necessary for the date-picking feature of the frontend dashboard page on the website.



Here, you can check your progress.
Pick a date to see your statistics for that specific day.

Figure 9.6: The statistics page on the website, automatically set to the date shown in bottom-left corner. The left-hand pie chart shows the three repetitions in blue and completed, as well as the one set. On the right, the rotation exercise's pie chart shows that for the set number one, only three out of five repetitions were completed.



Contraction progress = 20%; EMG value = 265; data = 0.294

Contraction progress = 40%; EMG value = 427; data = 0.475

Contraction progress = 60%; EMG value = 786; data = 0.874

Contraction progress = 80%; EMG value = 214; data = 0.237

Contraction progress = 100%; EMG value = 100; data = 0.111

Figure 9.7: a.: first contraction graph labeled at 0%, 20%, 40%, 60%, 80% and 100% execution progress. b.: hand visualisation at 20%, EMG value recorded as 265 and MATLAB data variable equal to 0.294. c.: hand visualisation at 40%, EMG value recorded as 427 and MATLAB data variable equal to 0.475. d.: hand visualisation at 60%, EMG value recorded as 786 and MATLAB data variable equal to 0.874. e.: hand visualisation at 80%, EMG value recorded as 214 and MATLAB data variable equal to 0.237. f.: hand visualisation at 100%, EMG value recorded as 100 and MATLAB data variable equal to 0.111.

Chapter 10

Conclusions

The project developed in this thesis work investigated the problem of non-compliance or low compliance to the rehabilitation and showed that technological solutions can help deal with it. The way this project wants to address the problem specifically is by providing a technological ecosystem to promote the patient's motivation, on the basis that it is strictly related to non-compliance, as demonstrated by research already. In addition, preliminary expert reviews supported the evidence for the lack-of-compliance issue in the specific application scenario of this project, which is Denmark. Moreover, such solutions have already been developed and tested successfully, proving that this approach to rehabilitation can be beneficial to the patients, as explained in 3.2.2.

On these grounds, a system composed of a physical prototype device, a mobile application and a website was designed, the RehTracker. The physical device was developed to collect data on two classes of exercises for the forearm area and transmit them to a mobile app via BLE, while showing the user real-time visual feedback of the single exercise repetition's execution. The app was then developed to read the data from the physical prototype and show the user - i.e. the patient - the up-to-date execution statistics, upon successful user sign-in and connection to the device. Meanwhile, a backend web service was designed and implemented to support data storage and user sign-in. Alongside, a mock-up frontend statistics dashboard was realised to illustrate an additional feature of an ideal complete ecosystem.

After development, it was shown that the RehTracker has the potential to be a real product and be used on real patients once addressed a few design issues, pointed out by the test users and the experts.

Beyond monitoring the rehabilitation, the concept of visual feedback was further investigated by creating the robotics visualisation, to provide a virtual representation of the muscles' contractions and support the idea of fostering motivation.

Overall, the proof of concept produced for this project is contextualised - but not strongly bonded - to the specific country it was developed in. Even so, similar technological solutions have been positively tested in other countries and RehTracker could be, then, easily adopted in other similar realities.

10.1 Answers to the research question

Non-compliance to the rehabilitation process has been shown to be highly influenced by the motivational factor that the patient experiences. As research shows, motivation can be increased by allowing more execution autonomy - in terms of easier time management - and feedback to make the patient aware of their progress.

On autonomy, RehTracker has been recognised to be non-invasive to wear by the experts and test users and the app-physical device system can be used at any desirable time. Thus, the author reckons that the autonomy component of intrinsic motivation is properly addressed by the project, in its current development stage.

At the same time, the visual feedback provided by the RehTracker's LEDs and the exercises' repetition counters on both app and website were perceived by the experts as addressing correctly the issue of lack of external regulation. Fostering feedback can positively increase extrinsic motivation to complete the prescribed exercises, according to research. Hence, the author also believes that RehTracker can improve patients' adherence by providing feedback in a visual form, additionally even more by incorporating the robotics visualisation's real-time visual feedback in the patient's rehabilitation routine.

10.2 Future works

The current ecosystem only sees the website as mock-up of a long-term statistics dashboard. So, firstly its development will have to be completed to provide a fully working and usable ecosystem. Moreover, the possibility should be added for the therapist to use the web platform for delivering instructions to the patient and/or see their day-to-day progress.

Issues pointed out on the mobile app will have to be resolved. Also, it will be beneficial to rework the physical prototype final design to gain even more compactness and make the pads/cables application easier and more intuitive.

Ultimately, to collect appropriate and relevant feedback, a test on real patients will have to be conducted upon applying the aforementioned corrections.

Bibliography

[1] AdvancerTechnologies. *3-lead Muscle / Electromyography Sensor for Microcontroller Applications - DATASHEET*. 2016. URL: https://cdn.sparkfun.com/assets/a/3/a/f/a/AT-04-001.pdf?_gl=1*1pxe1ry*_ga*MjMwODMxNDUxLjE2NDU2MTU2MTE.%20*_ga_T369JS7J9N*MTY5NTIyNTcxOS4%20xLjEuMTY5NTIyNTc0Ni4zMy4wLjA..

[2] Nefeli Alushi. *Accessing Accelerometer Data on Nano 33 BLE*. URL: https://docs.arduino.cc/tutorials/nano-33-ble/imu-accelerometer.

[3] Arduino. *Adafruit NeoPixel*. URL: https://www.arduino.cc/reference/en/libraries/adafruit-neopixel/.

[4] Arduino. *Arduino Nano 33 BLE*. URL: https://docs.arduino.cc/hardware/nano-33-ble.

[5] Arduino. *Arduino Nano 33 BLE Sense - EDITED*. URL: https://store.arduino.cc/products/arduino-nano-33-ble-sense.

[6] Arduino. *Arduino UNO R3*. URL: https://docs.arduino.cc/hardware/uno-rev3.

[7] Arduino. *Arduino_LSM9DS1*. URL: https://www.arduino.cc/reference/en/libraries/arduino_lsm9ds1/.

[8] Arduino. *ArduinoBLE*. URL: https://www.arduino.cc/reference/en/libraries/arduinoble/.

[9] Arduino. *Getting Started with Arduino IDE 2*. URL: https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2.

[10] Arduino. *loop()*. URL: https://www.arduino.cc/reference/en/language/structure/sketch/loop.

[11] Arduino. *setup()*. URL: https://www.arduino.cc/reference/en/language/structure/sketch/setup.

[12] Nordic Semiconductor ASA. *nRF Connect for Mobile*. URL: `https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-mobile`.

[13] Auth0. *Introduction to JSON Web Tokens*. URL: `https://jwt.io/introduction`.

[14] Auth0. *jsonwebtoken*. URL: `https://www.npmjs.com/package/jsonwebtoken`.

[15] Automattic. *mongoose*. URL: `https://www.npmjs.com/package/mongoose`.

[16] Jens Baes-Jørgensen. *Teknologiske løsninger øger motivationen til at genoptræne [Technological solutions increase motivation to rehabilitate]*. Danish. KL. Jan. 2020. URL: `https://www.kl.dk/nyheder/momentum/2020/nr-2/teknologiske-loesninger-oeger-motivationen-til-at-genoptraene/#:~:text=Kommunerne%5C%20vurderer%5C%20da%5C%20ogs%5C%C3%5C%A5%5C%20i,det%5C%20n%5C%C3%5C%A6re%5C%22%5C%20og%5C%20ikke%5C%20skal` (visited on 06/01/2022).

[17] B.R. Brewer, R. Klatzky, and Y. Matsuoka. "Initial Therapeutic Results of Visual Feedback Manipulation in Robotic Rehabilitation". In: *2006 International Workshop on Virtual Rehabilitation* (2006), pp. 160–166. DOI: `10.1109/IWVR.2006.1707546`. URL: `https://ieeexplore.ieee.org/abstract/document/1707546`.

[18] Rocco Salvatore Calabrò et al. "The role of virtual reality in improving motor performance as revealed by EEG: a randomized clinical trial". In: *Journal of NeuroEngineering and Rehabilitation* 14 (1 2017), pp. 53–. DOI: `10.1186/s12984-017-0268-4`. URL: `https://doi.org/10.1186/s12984-017-0268-4`.

[19] Embedded Centric. "Lesson 2 – BLE profiles, services, characteristics, device roles and network topology". In: (). URL: `https://embeddedcentric.com/lesson-2-ble-profiles-services-characteristics-device-roles-and-network-topology/`.

[20] Fatima Chaudhry et al. "Anatomy, Shoulder and Upper Limb, Forearm Compartments". In: *StatPearls* (July 2023). URL: `https://www.ncbi.nlm.nih.gov/books/NBK539784/`.

[21] Alarcos Cieza et al. "Global estimates of the need for rehabilitation based on the Global Burden of Disease study 2019: a systematic analysis for the Global Burden of Disease Study 2019". In: *www.thelancet.com* 396 (Dec. 2020). DOI: `https://doi.org/10.1016/S0140-6736(20)32340-0`.

[22] Pietro Cipresso et al. "The Past, Present, and Future of Virtual and Augmented Reality Research: A Network and Cluster Analysis of the Literature". In: *Frontiers in psychology* (2018). DOI: `10.3389/fpsyg.2018.02086`. URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6232426/`.

[23] MDN Contributors. "Express/Node introduction". In: *MDN Web Docs* (2023). URL: `https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction#introducing_express`.

[24] MDN Contributors. "Using HTTP cookies". In: *MDN Web Docs* (2023). URL: `https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies#restrict_access_to_cookies`.

[25] dart.dev. *http*. URL: `https://pub.dev/packages/http`.

[26] Paul DeMarco. *flutter_blue*. URL: `https://pub.dev/packages/flutter_blue`.

[27] Ihor Demedyuk and Nazar Tsybulskyi. "Flutter vs Native vs React-Native: Examining performance". In: *inVerita* (2020). URL: `https://inveritasoft.com/blog/flutter-vs-native-vs-react-native-examining-performance`.

[28] SparkFun Electronics. *Disposable Surface EMG/ECG/EKG Electrode - 24mm (10 pack) - WEBSHOP*. URL: `https://www.sparkfun.com/products/12969`.

[29] SparkFun Electronics. *MyoWare Cable Shield - WEBSHOP*. URL: `https://www.sparkfun.com/products/retired/13687`.

[30] SparkFun Electronics. *MyoWare Muscle Sensor - WEBSHOP*. URL: `https://www.sparkfun.com/products/retired/13723`.

[31] SparkFun Electronics. *NeoPixel Stick - 8 x WS2812 5050 RGB LED - WEBSHOP*. URL: `https://www.sparkfun.com/products/12661`.

[32] SparkFun Electronics. *Sensor Cable - Electrode Pads (3 connector) - WEBSHOP*. URL: `https://www.sparkfun.com/products/12970`.

[33] Flutter. *Dart overview*. URL: `https://dart.dev/overview`.

[34] Flutter. *Flutter architectural overview*. URL: `https://docs.flutter.dev/resources/architectural-overview`.

[35] BJ Fogg. *A Behavior Model for Persuasive Design*. Stanford University, Behavior Design Lab. 2009. URL: `https://behaviordesign.stanford.edu/resources/fogg-behavior-model#:~:text=The%5C%20Fogg%5C%20Behavior%5C%20Model%5C%20shows,to%5C%20understand%5C%20behavior%5C%20in%5C%20general`.

[36] OpenJS Foundation. *About Node.js*. URL: `https://nodejs.org/en/about`.

[37] Cornelia Győrödi and Robert Gyorodi. "A Comparative Study of Relational and Non-Relational Database Models in a Web-Based Application". In: *International Journal of Advanced Computer Science and Applications* (Nov. 2015). URL: `https://www.researchgate.net/publication/285752873_A_Comparative_Study_of_Relational_and_Non-Relational_Database_Models_in_a_Web-_Based_Application`.

[38] Jan Hartvigsen, Christian Mølgaard, and Karen Søg. *Forebyggelsen Af Skader Og Sygdomme I Muskler Og Led [Prevention of Muscle and Joint Injuries and Diseases]*. Danish. Vidensråd for forebyggelse. 2015. URL: `http://www.vidensraad.dk/sites/default/files/vidensrad_for_forebyggelse_forebyggelse_af_skader_og_sygdomme_i_muskler_og_led_2015.pdf`.

[39] Teresa Holmberg, Mikala Josefine Poulsen, and Michael Davidsen. *Muskel- og skeletlidelser i Danmark [Musculoskeletal disorders in Denmark]*. Danish. Statens Institut for Folkesundhed, Syddansk Universitet. 2015. URL: `www.si-folkesundhed.dk` (visited on 10/12/2022).

[40] W B van den Hout. "The value of productivity: human-capital versus friction-cost method". In: *Annals of the Rheumatic Diseases* 69.Suppl 1 (2010), pp. i89–i91. ISSN: 0003-4967. DOI: `10.1136/ard.2009.117150`. eprint: `https://ard.bmj.com/content/69/Suppl_1/i89.full.pdf`. URL: `https://ard.bmj.com/content/69/Suppl_1/i89`.

[41] Bluetooth SIG Inc. "Bluetooth Technology Overview". In: *Bluetooth* (2023). URL: `https://www.bluetooth.com/learn-about-bluetooth/tech-overview/`.

[42] Meta Platforms Inc. *React - A JavaScript library for building user interfaces*. URL: `https://legacy.reactjs.org/`.

[43] Salesforce Inc. *What is Heroku?* URL: `https://www.heroku.com/what`.

[44] Joseph Jobit. *Types of Arduino Boards – Quick Comparison on Specification and Features*. Circuit Digest. 2022. URL: `https://circuitdigest.com/article/different-types-of-arduino-boards`.

[45] Maja Bæksgaard Jørgensen, Michael Davidsen, and Janne Schurmann Tolstrup. *De samfundsmæssige omkostninger ved muskel- og skeletlidelser i Danmark [The societal costs of musculoskeletal disorders in Denmark]*. Danish. Statens Institut for Folkesundhed, Syddansk Universitet. 2017. URL: `https://`

www.sdu.dk/da/sif/rapporter/2017/de_samfundsmaessige_omkostninger_ved_muskel_og_skeletlidelser_i_danmark (visited on 10/12/2022).

[46] Roberto Meattini et al. "An sEMG-Based Human–Robot Interface for Robotic Hands Using Machine Learning and Synergies". In: *IEEE Transactions on Components, Packaging, and Manufacturing Technology* 8.7 (July 2018). DOI: 10.1109/TCPMT.2018.2799987. URL: https://ieeexplore.ieee.org/document/8301546.

[47] Claudio Melchiorri et al. "Development of the UB Hand IV: Overview of Design Solutions and Enabling Technologies". In: *IEEE Robotics & Automation Magazine* 20.3 (2013), pp. 72–81. DOI: 10.1109/MRA.2012.2225471.

[48] Microsoft. *TypeScript is JavaScript with syntax for types.* URL: https://www.typescriptlang.org/.

[49] Microsoft. *Visual Studio Code.* URL: https://code.visualstudio.com/.

[50] MongoDB. *Connect to Your Cluster.* URL: https://www.mongodb.com/docs/atlas/tutorial/connect-to-your-cluster/#connect-with-the-sample-application-3.

[51] MongoDB. *Get Started with Atlas.* URL: https://www.mongodb.com/docs/atlas/.

[52] What is MongoDB? *Get Started with Atlas.* URL: https://www.mongodb.com/what-is-mongodb.

[53] JD Nguyen and H. Duong. "Anatomy, Shoulder and Upper Limb, Hand Long Flexor Tendons and Sheaths". In: *StatPearls* (Aug. 2023). URL: https://www.ncbi.nlm.nih.gov/books/NBK546607/.

[54] World Health Organization. "ADHERENCE TO LONG-TERM THERAPIES : evidence for action". In: (2003). URL: http://apps.who.int/iris/bitstream/handle/10665/42682/9241545992.pdf.

[55] World Health Organization. *Rehabilitation.* World Health Organization. Jan. 2023. URL: https://www.who.int/news-room/fact-sheets/detail/rehabilitation (visited on 05/06/2023).

[56] Filippo Paganelli. *RehTracker_arduino-code - GITHUB.* URL: https://github.com/FilippoPaganelli/RehTracker_arduino-code.

[57] Filippo Paganelli. *RehTracker_full-stack-webapp - GITHUB.* URL: https://github.com/FilippoPaganelli/RehTracker_full-stack-webapp.

[58] Filippo Paganelli. *RehTracker_mobile-app - GITHUB*. URL: `https://github.com/FilippoPaganelli/RehTracker_mobile-app`.

[59] Filippo Paganelli. *RehTracker_robotic-demo - GITHUB*. URL: `https://github.com/FilippoPaganelli/RehTracker_robotic-demo`.

[60] Filippo Paganelli. *Robotics visualisation video demo*. URL: `https://www.youtube.com/shorts/xnoz1dpzcbo`.

[61] Jaydeep Patadiya. "Node.js vs Other Backend Technologies: Which is Right for Your Business". In: *Radix* (2023). URL: `https://radixweb.com/blog/node-js-vs-other-backend-technologies`.

[62] Shyamal Patel et al. "A review of wearable sensors and systems with application in rehabilitation". In: *Journal of NeuroEngineering and Rehabilitation* 9 (2012). ISSN: 1743-0003. DOI: `10.1186/1743-0003-9-21`. URL: `https://doi.org/10.1186/1743-0003-9-21`.

[63] Bruce Phillips. "Staying Power: Rethinking Feedback to Keep Players in the Game". In: *Game Developer* (Oct. 2009). URL: `https://www.gamedeveloper.com/design/staying-power-rethinking-feedback-to-keep-players-in-the-game`.

[64] CS Rand. "Measuring adherence with therapy for chronic diseases: implications for the treatment of heterozygous familial hypercholesterolemia." In: *The American journal of cardiology* (1993). DOI: `10.1016/0002-9149(93)90014-4`. URL: `https://pubmed.ncbi.nlm.nih.gov/8213501`.

[65] recharts. *recharts*. URL: `https://www.npmjs.com/package/recharts`.

[66] Richard M. Ryan and Edward L. Deci. "Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions". In: *Contemporary Educational Psychology* 25 (2000), pp. 54–67. DOI: `10.1006/ceps.1999.1020`. URL: `http://www.idealibrary.com/`.

[67] Privender Saini et al. "Philips stroke rehabilitation exerciser: a usability test". In: (Apr. 2008), pp. 116–122.

[68] H Dale Schunk. *Motivation in Education: Theory, Research, and Applications*. 4th ed. Pearson, 2013.

[69] James Shore and Shane Warden. *The Art of Agile Development*. 1st ed. O'Reilly Media, Inc., 2007, pp. 31–50. ISBN: 978-0-596-52767-9.

[70] Shinde Siddhesh. “What are the Key Pros and Cons of the Arduino Programming Language?” In: *Emeritus* (2023). URL: `https://emeritus.org/blog/coding-arduino-programming-language/`.

[71] Janet M. Six. “Usability Testing Versus Expert Reviews”. In: *UXmatters* (Oct. 2009). URL: `https://www.uxmatters.com/mt/archives/2009/10/usability-testing-versus-expert-reviews.php`.

[72] Emmy Sluijs, Gerjo Kok, and Jouke Zee. “Correlates of Exercise Compliance in Physical Therapy”. In: *Physical therapy* 73 (Dec. 1993), 771–82, discussion 783. DOI: `10.1093/ptj/73.11.771`.

[73] Remix Software. *react-router-dom*. URL: `https://www.npmjs.com/package/react-router-dom`.

[74] DJ Steele, TC Jackson, and MC Gutmann. “Have you been taking your pills? The adherence-monitoring sequence in the medical interview.” In: *The Journal of family practice* (Mar. 1990).

[75] steenbakker.dev. *flutter_secure_storage*. URL: `https://pub.dev/packages/flutter_secure_storage`.

[76] American Society for Surgery of the Hand. *Muscles*. URL: `https://www.assh.org/handcare/safety/muscles#Wrist%20Forearm`.

[77] The MathWorks, Inc. *Bluetooth Low Energy Communication*. URL: `https://nl.mathworks.com/help/matlab/bluetooth-low-energy-communication.html`.

[78] The MathWorks, Inc. *MATLAB*. URL: `https://nl.mathworks.com/products/matlab.html`.

[79] The MathWorks, Inc. *Simulink*. URL: `https://nl.mathworks.com/products/simulink.html`.

[80] Josef Wiemeyer and Annika Kliem. “Serious games in prevention and rehabilitation-a new panacea for elderly people?” In: *Eur Rev Aging Phys Act.* 9.1 (2012), pp. 41–50. DOI: `10.1007/s11556-011-0093-x`. URL: `https://eurapa.biomedcentral.com/articles/10.1007/s11556-011-0093-x`.

Appendix A

Preliminary expert review

Pre-development questions:

- **Q1:** Do you feel that or know if your patients are following the prescribed exercise regimens when they are not with you?
  - **R1:** It varies a lot. I feel like some patients do, while some patients don't. But there is definitely a difference between patients that keep doing exercises at home and those who don't.
  - **R2:** In the beginning of the injury, I feel like they do it a lot, but that might be because we see each other more often. Later in the rehabilitation, near the end, they do it less, I think.
  - **Q1.2:** Why do you think that is?
    - **R1.2:** There could be a lot of reasons for this. Maybe they are doing it but just not properly. Maybe even laziness. They must think they're doing fine, or that their session with me is enough already.
    - **R2.2:** In the beginning of the rehabilitation period, we see the patients more often, depending on the injury. But in the end, they might think they already recovered from the injury, while some patients actually didn't, which can enhance the possibility of consequences in the future.
- **Q2:** From 1 to 7, where 7 is a perfect rehabilitation period, how do you feel like your average patient is doing?
  - **R1:** Personally, I would assume that it is about 5 out of 7.

  - **R2:** I feel like it really depends on the injury. If the injury is really bad, I would think it is between 6 and 7. And if the injury is minor, such as a broken finger, it would be between 3 and 4, from my viewpoint at least.

- **Q3:** Usually, when a patient gets their plaster off after braking a hand or arm, they would get a piece of paper that needed exercises to get a full recovery. Do you sometimes give your patients supporting tools to help them with the rehabilitation? If so, then what do you give them?

  - **R1:** Answer not given.
  - **R2:** It depends on the injury. Sometimes we create a specific plan for that individual. We are planning based on their needs, and then we give them something they can use at home, or something we recommend them to buy. But if the injury is very common, a folder with guidance is common.

Appendix B

Late expert review

Post-development questions:

- **Q1:** Can you tell how it is to have the prototype on? How does it feel?
  - **R1:** It doesn't weigh that much in total, which is great. Felt extremely sensitive (refers to the LED response to their forearm movement). The lights are a very good addition to it. So, they can see how good they're doing.
  - **R2:** It did not weigh a lot, which is very important. The wires were a little long. The lights are very sensitive even though I'm not contracting any muscles (refers to the LEDs being too sensitive to small-to-null movements).
- **Q2:** Now that we've helped you put it on, can you put it on by yourself? How does that feel?
  - **R1:** If there could be any more obvious symbol or something that indicates which degree I should put it on the wrist, it would help a lot.
  - **R2:** The pads are probably the most difficult ones. They were already on in this case but it could be a problem for the patient that doesn't know exactly where to put them on the forearm. They are also kind of difficult to click onto the wires.
- **Q3:** What do you think about the feedback in form of lights?

  - **R1:** I think it is really helpful from the patient point of view, to see if they did it (refers to the exercise) correct and if they did the contraction enough.
  - **R2:** It's a really good thing. To confirm if you did an exercise sufficiently is essential, both for the patient and us. To see what you're doing in the same moment that you're doing it is great.

- **Q4:** Do you feel that the exercises in the leaflet (referring to figure 3.2) could be replicated properly? Anything you would add?

  - **R1:** I don't have anything specific to add at the moment. But being early in the process (refers to the RehTracker development) I think it won't be complicated to add exercises in the future.
  - **R2:** It could replicate the exercises pretty good. I would maybe add some more exercises that cover other muscles. It could be possible to cover more exercises.

- **Q5:** Do you think this artifact has the potential to be used as an instrument for the rehabilitation process? If so, how do you think it would affect the patient's rehabilitation?

  - **R1:** If it works as intended, then I could see it as a thing that could help both us and the patients, then I could see how they are doing at home. I think it would be nice for them to see their progress, especially while doing the exercises. It would be good to know (for the patients) if they're making it correctly and be confident.
  - **R2:** Yes, I definitely think that is possible, if the patient gives consent to us to monitor them. I think it would remove a lot of doubt about executing the exercise correctly.

- **Q6:** Would you be comfortable giving this product to your patients?

  - **R1:** To be honest, I have a hard time visualising it at the moment, but I would love to try it.
  - **R2:** If the final product is how I envision it to be out from your prototype and additional explanations, then I would ask some of my patients if they would like it. And if the functionalities have been tested properly and

documented that everything is working fine as intended, I would give it to my patients, yes.

Appendix C

User testing

User-test questions:

- **Q1:** Can you tell your initial thoughts on having the prototype on? How does it feel?
  - **R1:** I feel powerful, I feel like I have control over something. It is very light and comfortable to wear. I also think it is very cool that I could register my movements inside the app. (Walked around while wearing it) It is not annoying when I walk or do other things, which is good. I think it is pretty cool in general, it makes total sense.
  - **R2:** I didn't understand how it works; it was always lighting up when I moved my arm. But when you said you wanted me to contract my muscle, it made sense. But it was lighting even though I didn't do much. It felt not too bad, it was fine. But I think it would be good for rehabilitation. But I don't get the application.
- **Q2:** Now that you've been helped putting it on one time, can you put it on by yourself? How does it feel?
  - **R1:** It was difficult to put on the pads. Otherwise, the wrist was easy now that you told me how to do it once.
  - **R2:** Very difficult to re-attach the pads (refers to snapping the wires on them).
- **Q3:** What do you think about the lights? Anything else you would add?

- **R1:** It's nice to see if you're doing it right (refers to the exercise).
- **R2:** I think it is very reassuring for the patients to see right away, if they are doing it correct. The light was nice, but I would love to see something more, like inside a game or something, where you get points or rewards for doing good. And maybe it would be good for the product, if you made people understand the importance of rehabilitation as you told me.