BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

PROCESSING AND VISUALIZATION OF DIAGNOSTIC DATA FROM A BIONIC UPPER LIMB PROSTHESIS

ZPRACOVÁNÍ A VIZUALIZACE DIAGNOSTICKÝCH DAT Z BIONICKÉ PROTÉZY HORNÍ KONČETINY

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **ONDŘEJ MAHDALÍK**
AUTOR PRÁCE

SUPERVISOR **Ing. JIŘÍ HYNEK, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

153807

Institut:	Department of Information Systems (DIFS)
Student:	**Mahdalík Ondřej**
Programme:	Information Technology
Title:	**Processing and Visualization of Diagnostic Data from a Bionic Upper Limb Prosthesis**
Category:	Information Systems
Academic year:	2023/24

Assignment:

1. Study the use of bionic prostheses. Learn about the principles of processing and evaluating data from bionic prostheses.
2. Study the principles of capturing, processing and visualization of time-stamped (time series) and healthcare data. Explore technologies and libraries for visualizing such data.
3. Analyze the requirements for extending the Z-Bionics Hub platform to capture, process and visualize data from an upper limb bionic prosthesis.
4. According to the results of the analysis, propose a suitable extension of the Z-Bionics Hub platform to provide information on the use of the bionic prosthesis.
5. Implement the proposed extension.
6. Test the usability of the solution in cooperation with Z-Bionics.

Literature:

- Basumatary, H., & Hazarika, S. M. (2020). State of the art in bionic hands. *IEEE Transactions on Human-Machine Systems*, *50*(2), 116-130.
- Saudabayev, A., & Varol, H. A. (2015). Sensors for robotic hands: A survey of state of the art. *IEEE Access*, *3*, 1765-1782.
- Triwiyanto, T., et al. (2021). State-of-the-art method in prosthetic hand design: A review. *Journal of Biomimetics, Biomaterials and Biomedical Engineering*, *50*, 15-24.
- Zuo, K. J., & Olson, J. L. (2014). The evolution of functional hand replacement: From iron prostheses to hand transplantation. *Plastic Surgery*, *22*(1), 44-51.

Requirements for the semestral defence:
Items 1 - 4.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Hynek Jiří, Ing., Ph.D.**
Head of Department:	Kolář Dušan, doc. Dr. Ing.
Beginning of work:	1.11.2023
Submission deadline:	16.5.2024
Approval date:	30.10.2023

Abstract

Bionic prostheses are commonly used to help patients perform everyday tasks more efficiently. They incorporate various methods to obtain input from the user and use electric motors to control fingers and/or other parts of the limb. Unfortunately, the rejection rates are averaging 44% [64], meaning almost half of the patients stop using the prosthesis and put it away. The Z-Arm prosthesis from the company Z-Bionics attempts to tackle the low acceptance rate by logging diagnostic data and helping the patient when improper usage or possible prosthesis defect is detected. However, since patient visits are not often enough to provide engineers with recent diagnostics, a remote way of obtaining them must be designed. A mobile application utilizing Bluetooth to communicate with the prosthesis has been implemented as a solution. An extension of an internal web application with new pages, API endpoints and a MongoDB database accompanies it. While the testing is still ongoing and results are yet to be measured, first impressions of staff and patients are positive, indicating a possible improvement in the comfort of prosthesis usage.

Abstrakt

Bionické protézy se běžně používají k tomu, aby pacientům pomohly efektivněji vykonávat každodenní činnosti. Zahrnují různé metody získávání vstupu od uživatele a používají elektromotory k ovládání prstů a/nebo jiných částí končetiny. Bohužel míra odmítnutí protézy dosahuje v průměru 44 % [64], což znamená, že téměř polovina pacientů přestane protézu používat. Protéza Z-Arm od společnosti Z-Bionics se pokouší řešit nízkou míru akceptace mimo jiné tím, že zaznamenává diagnostické údaje a pomáhá pacientovi, když detekuje nesprávné používání nebo možnou vadu protézy. Protože však návštěvy pacientů nejsou dostatečně časté, aby specialisté získali aktuální diagnostické údaje, je třeba navrhnout vhodný způsob jejich získávání na dálku. Jako řešení byla implementována mobilní aplikace využívající Bluetooth ke komunikaci s protézou doprovázená rozšířením interní webové aplikace o nové stránky, koncové body API a databázi MongoDB. Ačkoli testování s pacienty stále probíhá, první dojmy zaměstnanců a pacientů jsou vesměs pozitivní a naznačují možné zlepšení komfortu při používání protézy.

Keywords

prosthesis, limb, amputation, heathcare, diagnostic data, mobile application, information system, dashboard, API, visualization, MAUI, Blazor, .NET, Z-Bionics

Klíčová slova

protéza, končetina, amputace, zdravotnictví, diagnostická data, mobilní aplikace, informační systém, API, vizualizace, MAUI, Blazor, .NET, Z-Bionics

Reference

MAHDALÍK, Ondřej. *Processing and Visualization of Diagnostic Data from a Bionic Upper Limb Prosthesis.* Brno, 2024. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Jiří Hynek, Ph.D.

Rozšířený abstrakt

Amputace končetiny má závažný dopad na každodenní život pacienta, který je mnohem větší, než si většina z nás dokáže představit. Amputace mohou být způsobeny vrozenými vadami, většina je ovšem zapříčiněna úrazem, například při práci se stroji nebo při dopravní autonehodě [66]. Protézy se k usnadnění běžných činností pacientů s amputací používají již po staletí, přičemž první známá protéza pochází již ze 3. století př. n. l. [22]. Největší rozšíření ovšem zaznamenaly po první světové válce, a to zejména kvůli větší pravděpodobností přežití díky lepší zdravotní péči. I přes všechen vývoj na poli bionických protéz, zejména v posledních dvou dekádách, míra přijetí protézy stále zůstává na velmi nízké úrovni, a to na pouhých 66 % [64].

Český startup Z-Bionics, který se specializuje na výrobu bionických protéz na míru, pracuje na způsobech, jak míru přijetí zvýšit. Jednou z cest, jak toho dosáhnout, je odhalit potencionální problémy s protézami nebo nevhodné používání dříve, než povedou k tomu, že pacient protézu odloží a už ji znovu nepoužije. Proto je protéza horní končetiny Z-Arm vybavena funkcí sběru diagnostických dat. Bohužel vzhledem k tomu, že návštěvy pacientů se konají zpravidla jen jednou za několik měsíců, získání a vyhodnocení diagnostických dat může nastat pozdě, a proto je třeba navrhnout a implementovat způsob průběžného získávání těchto dat vzdáleně.

Při výběru nejlepšího způsobu získávání diagnostických dat bylo zváženo několik řešení. Patřilo mezi ně přidání modulu pro IoT na PCB desku protézy, vývoj chytré USB C nabíječky, pomocí které by se protéza nabíjela a která by byla připojená k internetu prostřednictvím Wi-Fi nebo mobilní aplikace využívající Bluetooth. Nakonec byla zvolena mobilní aplikace, protože nevyžadovala hardwarové změny na protéze ani vývoj nového hardwaru, jako v případě chytré nabíječky. Toto řešení by tudíž šlo použít i pro již vyrobené protézy a vyžadovalo by pouze aktualizaci firmwaru. Dalším benefitem mobilní aplikace, který se nakonec ukázal pro pacienta v krátkodobém horizontu dokonce nejpřínosnějším, jsou další funkce, které mobilní aplikace může obsahovat, zejména konfigurace protézy.

Dalším prvkem bylo API rozhraní, do kterého aplikace odesílá získaná diagnostická data. Vzhledem k tomu, že jsme již měli naši interní aplikaci Z-Bionics Hub, kterou jsme používali hlavně pro inventuru zásob, rozhodl jsem se ji rozšířit o další funkce, které by umožnily zpracování a vizualizaci těchto dat.

Výsledkem je mobilní aplikace napsaná nad frameworkem .NET MAUI, která komunikuje přes Bluetooth s protézou a využívá CBOR pro snížení velikosti zpráv, API rozhraní a rozšíření webové aplikace Z-Bionics Hub o nové stránky pro správu protéz a vizualizaci diagnostických dat. Mobilní aplikace umožňuje kromě sběru diagnostických dat i konfiguraci protézy. Nastavovat lze chování protézy v jednotlivých režimech – tzv. gripech, ukládání pojmenovaných předvoleb, nastavení citlivosti senzorů, jasu a automatického vypínání. Dále aplikace duplikuje možnosti běžně dostupné stisknutím tlačítek na protéze (spuštění/vypnutí, změna režimu, pohyb zápěstí nebo loktu). API rozhraní je zabezpečeno pomocí JSON Web Tokenu (JWT) a umožňuje nahrávání diagnostických dat do databáze. Backend byl rozšířen o druhou databázi dedikovanou pro diagnostická data – MongoDB. Kolekce v databázi jsou nastaveny v režimu TimeSeries. Webová aplikace následně tato data vizualizuje prostřednictvím grafů.

Testování některých částí aplikace je realizováno formou automatizovaných unit a end-to-end testů. Mobilní aplikace a webové vizualizace byly testovány ručně. Na testování se kromě autora podíleli další zaměstnanci Z-Bionics a bylo zahájeno testování s pacienty, kterého se v době psaní této práce účastní jeden pacient, v brzké době se nicméně očekává zapojení dalších pacientů. Testování s pacienty potrvá několik měsíců, tudíž zatím nelze uvádět konkrétní závěry, ovšem první dojmy pacientů ohledně mobilní aplikace jsou vesměs pozitivní, což naznačuje možné zlepšení komfortu při používání protézy.

Processing and Visualization of Diagnostic Data from a Bionic Upper Limb Prosthesis

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Mr. Ing. Jiří Hynek, Ph.d. The supplementary information was provided by company Z-Bionics s.r.o. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
Ondřej Mahdalík
May 16, 2024

Acknowledgements

I would like to thank to Mr. Ing. Jiří Hynek, Ph.D. for all the provided feedback and Z-Bionics employees for the given support.

Contents

Chapter 1

Introduction

Having a limb amputated is a life-changing event with consequences more significant than most of us imagine. Limb amputation affects millions of people worldwide. In the United States alone, 550,000 Americans were living with an upper extremity amputation in 2005 [71]. Bionic prostheses are commonly used to help patients perform everyday tasks more easily. They have been used for centuries; however, despite the significant development after the 1st and 2nd World Wars, upper limb prostheses have a notably low acceptance rate, averaging 66% [71].

One of the ways to increase the acceptance rate is to gain an insight into how each patient uses their prosthesis and to monitor the prosthesis' state, detecting and dealing with possible issues before they result in the patient's dissatisfaction, which might ultimately result in the patient putting the prosthesis away and never using it again. Since patient visits only occur every few months, the sole diagnostics collection is not sufficient, and a way of obtaining them remotely needs to be designed and implemented.

The aim of this thesis is, therefore, to propose a solution for obtaining the diagnostic data, implement it and test under real-life conditions. To achieve this, I have collaborated with a company called Z-Bionics, a Czech startup that specializes in making custom-made bionic prostheses and allows me to implement the solution in a real product and test it with real devices and patients to ensure that it brings expected benefits.

The thesis begins with research into the history of bionic prostheses, summarizes the current state with emphasis on myosignal acquisition and provides a brief overview of the Z-Bionics Z-Arm prosthesis in Chapter 2. Chapter 3 continues with research into the processing and visualization of healthcare time series data and related GDPR specifics and also describes ways of visualizing obtained data. Target groups and their needs are analyzed in Chapter 4, and so is the current state of the Z-Arm prosthesis and relevant Z-Bionics tools. A solution consisting of a mobile application communication with the prosthesis via Bluetooth and changes to the Z-Bionics Hub, its databases and API are proposed in Chapter 5. Chapter 6 describes the implementation process, used frameworks and libraries, and any noteworthy deviations from the proposed solution. Implemented changes were tested using automated tests, manually by the author, and by Z-Bionics employees and patients who use the Z-Arm prosthesis in Chapter 7.

Chapter 2

Bionic Prostheses

A bionic prosthesis is an artificial limb that closely mimics a real limb's natural movements and functions [2]. It incorporates a range of sensors and motors that enable it to respond to the user's commands. Most common prostheses are of lower limbs due to the considerably more significant number of patients suffering from lower limb amputation [71] and their relative simplicity compared to upper arm prostheses. This thesis will, however, focus on the upper arm's bionic prostheses.

2.1 History of Bionic Prostheses

The first known prostheses date to the 3rd Century BC, when a Roman general, Marcus Sergius, lost a hand in the Second Punic War and had an iron prosthesis made for him, which enabled him to return to battle [22]. Early prostheses were relatively simple, usually made of wood or iron, optionally covered with leather, and often designed for a particular use (holding a shield, for Example). The primary cause of the prolonged development of prostheses in Ancient history and the Medieval Ages was little need for them. Due to the state of Medicine, amputation of a limb used to be, in most cases, fatal, with the cause of death being an infection or haemorrhage that followed the initial injury [22].

A more advanced example from the 16th Century was a set of prostheses made for a German mercenary called Goetz Von Berlichingen, one of which offered independent movement of phalanges and opposition of thumbs, allowing the knight to hold the horse's rein [63]. The demand for prostheses spiked after World War I and II due to a large number of veterans suffering devastating extremity injuries [83]. Upper limb prostheses created in the early and mid 20th Century mainly were not built for cosmetic purposes but were instead designed for practical use, often consisting of a base part that was attached to the body and that contained a universal socket and multiple attachments tailored for a specific use that could be attached to the socket [49].

The first myoelectric prosthesis was developed by Reinhold Reiter in 1948 [83]. Electric motors actuated it, and the myosignal was obtained using sEMG signals. sEMG sensors were placed on the remaining muscles (usually biceps or triceps), and their tightening was used for prosthesis control. The prosthesis, however, was not portable and was designed to be used as a worktable tool plugged into a power outlet. Unfortunately, this prosthesis was not clinically or commercially adopted [65]. Despite providing more intuitive controls over body-powered prostheses, early myoelectric prostheses had significant drawbacks that severely affected their acceptance rate by patients. These include being considerably heavier than the natural limb, providing no feedback, and forcing the patient to visually check the prosthesis. Furthermore, they were prone to accidental activation that could result in dropping a grasped object. The majority of these drawbacks persist even today [69].

In the early 2000s, targeted muscle reinnervation (TMR) and targeted sensory reinnervation (TSR) attempted to tackle the lack of feedback [39]. It is a surgical procedure that consists of denervating a small area of skin and then reinnervating it with fibres of remaining hand nerves [28]. When such an area is touched, it provides the patient with a sense of a missing limb being touched [55]. This is then used to provide feedback to the user, allowing for distinguishing multiple levels of force, grasped object size, and density [39].

In recent years, other ways of improving the prosthesis wear comfort and use intuitiveness are being tested and implemented in new prostheses. These include using lightweight 3D-printed materials for reduced weight [12], pattern-recognition-based myoelectric control [11], and can also incorporate neural networks [80].

2.2 Current State of Bionic Prostheses

Today's upper limb bionic prostheses feature advanced myosignal acquisition and electromechanical control of fingers. Both custom-made and serially produced prostheses are available, offering a range of biosignal acquisition methods and supplementary sensors for improved command recognition [72]. 3D printing and composite materials are commonly used during manufacturing, allowing for considerably lighter prostheses that improve user comfort by weighting similarly to a natural arm. Some experiments incorporate a neuromusculoskeletal interface, which is directly connected to the bone using an osseointegrated implant to solve issues related to traditional socket suspension. Implanted electrodes provide bi-directional communication with the prosthesis [57].

Bionic prosthesis control methods can be split into neuromuscular (myosignals), neural, and brain signals. There are also other, rather unusual, ways of control (for Example, using IMU (Inertial measurement unit) sensors, Voice control, etc.). The most commonly used method is the myosignal acquisition, which has been used since Reinhold Reiter's invention as described in 2.1 [48]. Figure 2.1 briefly overviews various techniques for controlling a bionic prosthesis. Section 2.2.1 describes some of the techniques depicted.

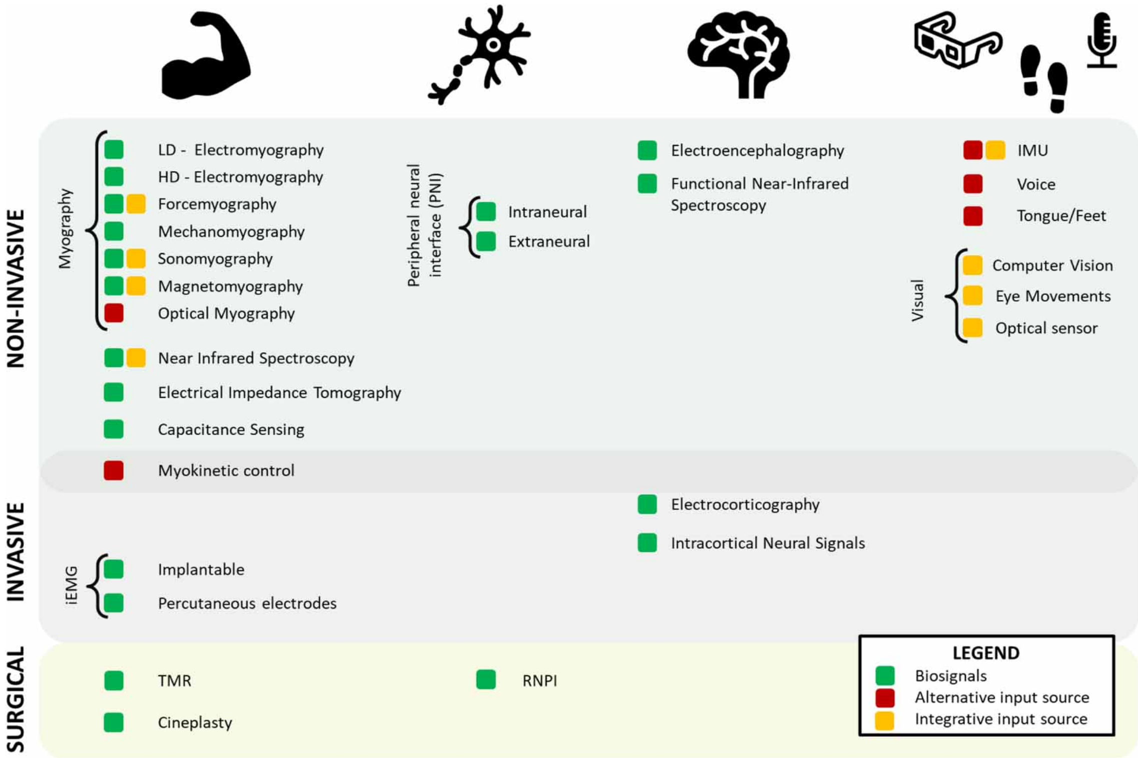

Figure 2.1: Overview of available signal acquisition methods[1] [48]

2.2.1 Myosignal Acquisition

Myosignal can be acquired in a number of ways, both invasive and non-invasive, each offering different advantages and disadvantages. Most important differences include signal sensitivity, muscle cross-talk (detecting activity of surrounding muscles that we do not want to measure), other forms of interference, wear comfort, long-term usage difficulties, etc. The oldest and, to this day, most commonly used myosignal acquisition method is using surface electromyography (sEMG) sensors [44]. An EMG sensor is a compact device with two or three electrodes and is non-invasively placed on the skin [43]. It captures the electric potential that muscle cells generate as they contract [58]. The sensor usually consists of two electrodes that are placed on the skin above the muscle that is being measured. The sensor then measures the voltage difference between electrodes [77]. Ideally, two sensors are placed on antagonist muscles, which increases accuracy and enables easier filtering of crosstalk and other interference [48].

EMG offers a comfortable way of measuring myosignal and is suitable for use in prosthetics thanks to its compact size and small energy requirements; however, it is important to recognize its drawbacks because they severely affect the accuracy of measurement. Due to the EMG sensor being placed on the skin, it suffers from skin impedance, sweat, and electrode shift [13]. Additionally, deep muscles cannot be reliably measured, which is especially problematic for amputees as their muscles on the residual limb degrade as a result of very little activity [48]. Commercially available EMG sensors include Myoscan SA9503M [8], Biometrics Ltd.'s SX230 [9], and Elemyo's MYO v1.4 [15]. They differ in dimensions,

[1]Image used with permission by Copyright Clearance Center, Inc. on behalf of the Rightsholder.

weight, electricity consumption, sensitivity, and the range of frequencies they are able to capture; however, considering that the captured signal requires additional processing and filtering in order to be usable regardless of the sensor used, the most important differences are the dimensions and weight because they affect both patient comfort and restrict the design of the socket and prosthesis [82].

A different approach that aims to obtain EMG signals more precisely and reliably is invasive electromyography (iEMG). Unlike sEMG, electrodes are implanted directly into muscles during a surgical procedure using either needles or wires as electrodes [24]. While the common sEMG drawbacks, such as muscle crosstalk, sweat, and electrode shift, are greatly mitigated, the need for surgery and the implants being connected to the prosthesis by a wire bring significant challenges, such as increased discomfort in the form of pain and a risk of infection and other complications related to the surgical procedure [48]. Therefore, iEMG is more suitable for short-term diagnosis and research, where precision is crucial, and infection risk resulting from daily usage is greatly reduced [24, 48].

As described in Section 2.1, another method that further exploits electromyography is targeted muscle reinnervation (TMR) and targeted sensory reinnervation (TSR). In TMR, the remaining limb nerves that originally controlled the muscles of an amputated limb are surgically redirected to nearby muscles. The nerves then grow into the muscles, and when the patient thinks about moving the missing limb, the signals from the brain are picked up by the reinnervated muscles [47]. Then, sEMG sensors obtain myosignal, which is then used for prosthesis control. The capabilities of TMR are very dependent on the individual patient. The TMR has the drawbacks of both sEMG and iEMG because surgery is necessary to perform the reinnervation [76]. On the other hand, TSR focuses on providing sensory feedback to the patient by restoring a sense of touch [55].

Other myosignal acquisition techniques include forcemyography (FMG), which uses force sensors that detect muscle contraction [19]; mechanomyography (MMG), which is based on detecting vibrations generated by contracted muscles [16]; or sonomyography, which detects changes in muscle thickness when contracted using ultrasound [54]. Alternatively, a bionic prosthesis can be controlled using other types of inputs, for Example, voice commands [7], accelerometers, gyroscopes, or magnetometers [40]. These input sources can be used as a primary control source; however, in most cases, they only supplement myosignals to enhance the intuitiveness and accuracy of the myosignal acquisition [7, 40].

Despite the aforementioned highly advanced methods, such as TMR and TSR, non-invasive sEMG sensors still prevail as the most common myosignal input method for prosthetics [44]. One of the prostheses using sEMG sensors to obtain myosignal is the Z-Bionics Z-Arm. The following text will focus on this prosthesis.

2.3 Z-Arm Prosthesis

The Z-Arm is a bionic upper arm prosthesis. It is manufactured individually for each customer and incorporates 3D scanning and printing for perfect fit. Figure 2.2 shows a render of a Mk3 version of the Z-Arm prosthesis, and Figure 2.3 captures a process of 3D scanning a patient's residual limb. A 3D scan of a healthy arm is captured as well and serves as a reference when tailoring the prosthesis to match the size of the healthy arm. The prosthesis is attached to the limb using a socket that is selected by a physician and custom-made by an orthopaedic company[2] [82]. The finished prosthesis mounted on a socket can be seen in Figure 2.4.

Figure 2.2: Z-Arm Mk3 prosthesis [82]

[2]The company is called „ortopedická protetika“ in Czech

Figure 2.3: 3D scanning a client [82]

Figure 2.4: Finished prosthesis (Z-Bionics Arm Mk2 model) worn by a client [82]

The myosignal acquisition is ideally achieved using a pair of sEMG sensors placed on antagonistic muscles. The currently used sensor is the Elemyo v1.4 (figure 2.5). Some patients, however, do not have a pair of antagonistic muscles left on their residual limb. In such a case, a single sEMG sensor must be used, reducing accuracy. The captured myosignal is then amplified and filtered. An example of the final signal used for control can be found in Figure 2.6. In addition to sEMG sensors, the prosthesis also features two gyroscopes that can be used to determine its position and react to myosignal input accordingly.

Figure 2.5: Elemyo v1.4 EMG sensor [15]

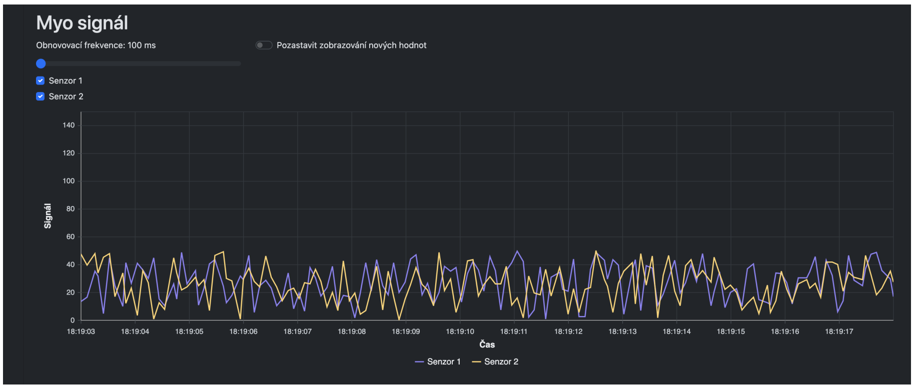

Figure 2.6: Example of captured myo signal displayed using an internal configurator utility [82]

2.3.1 Motor Control

Brushed DC electric motors control fingers. Each finger has its motor, allowing for individual control, which is especially useful when gripping objects. The finger's position is calculated using a potentiometer by calculating the ratio between maximum and minimum extension of the finger. The smoothness of the movement is ensured thanks to feedforward control, which allows for precise control of the motor's speed. A crucial part of the finger's movement is the gripping force, which allows for holding objects. The first step is to detect the object, which is done by observing the feedforward control's interventions. In the future, grip detection will be done by observing the acceleration and current draw of the motor, allowing for faster detection while reducing the number of false positives.

2.3.2 Diagnostic Data

To improve the client's comfort by making the prosthesis as intuitive as possible, especially when it comes to the handling of myosignals, and also to detect issues, preferably before they result in permanent damage to the prosthesis, the firmware in the prosthesis collects and stores diagnostic data and is equipped with small storage which can be used to store the captured data temporarily.

Chapter 3

Processing and Visualization of Healthcare Data

Working with medical data, especially those containing sensitive patient information, is very distinct from other disciplines. Challenges such as privacy, legal and ethical issues, missing values, and bias control arise and must be considered at all times while obtaining or analyzing such data and when storing it.

3.1 Security and Privacy Aspects

Recognizing security challenges associated with processing personal data, especially patients' sensitive medical records, is crucial in keeping such information safe. Great care should be put into the security of all systems that will have access to sensitive information. In order to implement a secure system, these basic goals need to be taken into consideration [67]:

- Confidentiality—data are inaccessible to individuals who are not authorized.
- Integrity—relevant information cannot be modified by unauthorized people.
- Availability—request data are obtainable when required by an authorized person.
- Authentication—being able to ensure that the person really is someone who they claim to be.
- Access control—being able to keep non-permitted users out of the system.
- Non-Repudiation—inability to dispute the authorship of a statement.

However, effort must also be put into a secure development and production environment. Whether it is making sure the CI/CD pipeline cannot be altered and malicious code is not injected when compiling code (as happened during 2020 United States federal government data breach [75]), or ensuring that the servers and software running on them are receiving security updates. Also, employees should only be given the permissions they need to perform their work, and proper authentication methods, such as multi-factor authentication, should be used. Lastly, recognizing the influence of human factors and educating both employees and clients on cybersecurity should not be underestimated [59].

Privacy is also an important topic to address. There are multiple ways of ensuring the privacy of patients is protected both ethically and per legal requirements [14]. Ways of protecting privacy include collecting only the data required, using data anonymization when possible, and deleting all data that is no longer needed [14, 60].

Due to the sensitive nature of medical data, handling it is bound by strict requirements and regulations, such as the General Data Protection Regulation (GDPR)[1] in Europe and the Health Insurance Portability and Accountability Act (HIPAA)[2] in the United States [36]. This thesis will mainly focus on GDPR-related challenges.

3.2 GDPR Specifics

Article 5.1-2 of GDPR defines seven data protection and accountability principles when processing personal data. These are lawfulness, fairness and transparency, purpose limitation, data minimization, accuracy, storage limitation, integrity and confidentiality, and accountability. The data subject (patient) has privacy rights that have to be considered: the right to be informed, the right of access, the right to rectification, the right to erasure, as well as other rights [17]. The *right to be informed* requires the data subject to be informed about the collection and use of the personal data as well as about the duration of storage, the rights of the data subject, their ability to withdraw their consent, and about their right to file a complaint with the authorities. The data subject has to be informed in a transparent, easily accessible, and understandable way [35]. The *right of access* ensures that the data subject has, upon request, the right to obtain information if any data related to them are being collected and processed, and if so, also at least the following [20]:

- processing purposes,
- categories of data concerned,
- recipients or categories of recipients to whom the data are disclosed,
- period for which the data is intended to be stored, or, if not possible, the criteria used to determine that period,
- existence of rights to rectify or to erase personal data or to restrict personal data processing,
- right to lodge a complaint with the supervisory authority.

The *right to rectification* allows data subjects to request correction of any inaccuracies in their personal data. The *right to erasure*, otherwise known as the right to be forgotten, provides data subjects with a right to have their personal data erased upon request, unless [20]:

- There are overriding legitimate grounds for the processing.
- The processing of personal data is necessary for exercising the right of freedom of expression and information; reasons of public interest in the area of public health; archiving purposes in the public interest, scientific or historical research purposes or statistical purposes; the establishment, exercise or defence of legal claims.

[1] https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex:32016R0679
[2] https://www.hhs.gov/hipaa/index.html

3.3 Time Series Data

Some healthcare data, such as patient vitals or medical device logs, are captured periodically and contain a timestamp, and thus can be considered as time series data.

Time series (TS) is a sequence of data captured in specific time intervals and, therefore, can be seen as a sequence of discrete-time data [79]. An example of TS data includes log files or data captured by IoT devices. TS data can be split into two basic categories [38]:

- Uni-variate TS—a single value corresponds to a timestamp.
- Multivariate TS—each timestamp is linked with an array (or vector) of real values.

The information obtained by examining the data can be used for future predictions. That is done by analyzing underlying patterns in the TS data [6].

Medical datasets are relatively small compared to other disciplines and often can be collected only in a non-reproducible way. This increases the risk of the data being affected by measurement errors, missing data, and patient bias [41].

3.3.1 Classification of Time Series Data

Classification of TS data is crucial in data analysis and machine learning. It involves labelling TS data based on their characteristics and patterns. This classification helps to identify different patterns in the data and provides a better understanding of the data for future predictions. It can also help make informed decisions and take appropriate actions based on the patterns observed in the data. [4]

Generally, classification algorithms can be split into feature-based, distance-based, and model-based methods [81]. The feature-based classification consists of extracting a set of features representing the global TS patterns [42] and then applying conventional classification methods [81]. In distance-based classification, distance metrics are used to determine class membership [4]. Finally, model-based classification presumes identical model generation for all class series and assigns new series to the best-fitting model's class [1].

3.3.2 Storage of Time Series Data

Compared to traditional datasets, TS datasets are relatively larger in size and are much more uniform [34]. TS data also don't require indefinite storage and can be aggregated or dropped altogether after a certain period. Such specific needs require well-optimized storage capable of storing large amounts of data and handling real-time data ingestion. It should also provide advanced features such as data compression, indexing, and aggregation to optimize performance and reduce storage costs [45]. The storage should be able to handle both univariate and multivariate TS data and support various data types and formats. Additionally, it should offer easy integration with other tools and platforms for data visualization, analysis, and machine learning. Finally, good storage for TS data should ensure data security, privacy, and compliance with relevant regulations and standards [5].

Relational databases

> A relational database is a (most commonly digital) database based on the relational model of data, as proposed by E. F. Codd in 1970. [78]

A relation database is based on the relation model, and it stores and manages data using a collection of tables with rows and columns. Each table represents a specific entity or relationship, and each row represents a specific instance of that entity or relationship. The columns in a table represent the attributes or properties of the entity or relationship. In a relational database, data is organized into tables with defined relationships between them, and data can be accessed and manipulated using SQL (Structured Query Language) commands [56]. As of 2023, relational databases remain the most common types of databases, with SQL databases taking four places in global use statistics [70].

TS data can be stored in a table with columns for timestamp and value, as well as additional columns for any other relevant metadata. TS data can also be stored across multiple tables, with each table representing a different time period or granularity. However, queries on large TS datasets can be slow and resource-intensive, particularly when querying across multiple tables; therefore, traditional relational databases are usually not a recommended choice for storing TS data [74]. A notable exception is PostgreSQL and its TimescaleDB extension.

TimescaleDB

TimescaleDB is an open-source PostgreSQL extension that adds support for efficient TS hosting [25]. It introduces hyper tables, which are PostgreSQL tables customized for the purpose of handling TS data. They automatically partition the data by time and optionally by space. Hypertables also create indexes on time and space if used for partitioning [73]. Due to being built upon PostgreSQL, a relation database, TimescaleDB fully supports SQL queries. At the same time, it has an expansion model comparable to NoSQL databases [68].

NoSQL Databases

With the continuous development of cloud computing and the Internet in general, a demand has been raised for databases capable of high concurrent reading and writing with low latency, efficient big data storage and access, high scalability and availability, and lower management and operational costs. As a response, new types of databases emerged. These new databases are referred to as „NoSQL“ databases, and as the name implies, they are generally very different from traditional relational databases [27]. Unlike relation databases, NoSQL databases do not support ACID transactions but instead use a softer consistency requirement called BASE (basically available, soft state, eventually consistent) [50], and implement the CAP theorem [23], which consists of:

- Consistency—each server should return the right response to each request.
- Availability—each request should eventually receive a response.
- Partition tolerance—servers can be partitioned into multiple groups that cannot communicate among them.

It is important to note that a distributed system cannot realistically meet all three needs simultaneously, but only two [27], and the purpose of the system is usually the key to determining which needs should be focused on. Despite NoSQL databases being considerably newer than relation databases, thanks to the high demand, we can nowadays find many different databases (described in, with each providing a different idea of storing and accessing data (as described in Section 3.3.2), the optimization for running in the cloud, or available client libraries.

Types of NoSQL Databases

NoSQL databases use different approaches to data representation models. These approaches can be split into four categories [53]: In *Key-Value oriented* architecture, each stored item has a corresponding key, and all queries on the data use it. Examples of Key-value-oriented databases include Oracle NoSQL and OrientDB. *Columns oriented* databases store data as columns instead of rows. Commonly used columns-oriented databases are Apache Cassandra and Google Bigtable. An alternative to column-oriented databases are *document-oriented* databases. They extend the key-value principle by considering the structure of the stored data as a document. The most common databases are MongoDB, Azure DocumentDB, and MarkLogic. The last category is *graph-oriented* databases model. It is designed to overcome problems associated with very complex connectivity links between data, which makes it suitable, for example, for social networks. Examples include Apache Giraph, Neo4j, and VelocityDB.

Some databases, like Amazon DynamoDB, InfluxDB, and Azure Cosmos DB, can use a combination of these types in order to serve a specific purpose [62]. This thesis will focus on the most commonly used NoSQL databases with dedicated support for TS data.

InfluxDB

InfluxDB is an open-source database designed to store and process TS data. It is developed by InfluxData. It is written in Go and provides a query language similar to SQL [21]. It offers official client libraries for almost all major high-level programming languages [33], which include C#, Go, Java, JavaScript, PHP, Python, etc.

An InfluxDB database always contains a time column, which stores timestamps associated with specific data [52]. Timestamps are stored in epoch nanosecond format. Another mandatory column is the measurement column. It describes the data and is stored in the form of a string. The measured values are stored as fields in the form of a key-value pair. They are not indexed. Additional metadata can be stored as tags, once again in the form of a key-value pair [31]. They are optional, indexed, and, when used properly, can improve the query performance [32]. InfluxDB allows configuring a retention policy and handles the deletion of old data without any user intervention [29]. Data processing and aggregation can be achieved using tasks—flux scripts that can be run periodically, query data, analyze or modify it, and write it back to the database. They can also be used to detect anomalies and, for example, send notifications about them [30].

MongoDB

MongoDB is an open-source document-oriented general-purpose database. It supports generic secondary indexes, provides an aggregation framework, and supports time-to-live collections for temporarily stored data, such as logs [10]. Version 5 introduced native support for TS data, called MongoDB TS Collections [3]. This collection type organizes writes in order to store data from the same sources together with other data points from a similar point in time [51]. MongoDB doesn't enforce any predefined schema, which includes document keys. This makes adding and removing fields as needed easier and can accelerate development by allowing developers to iterate quickly [10]. MongoDB is also well-optimized for cloud hosting, and the company even offers its own cloud solution called MongoDB Atlas [61].

3.3.3 Visualization of Time Series Data

Visualization of time series data can help understand it faster and notice patterns that might not be otherwise visible [18]. Various visualization techniques can be utilized in order to visualize the data effectively [26]. They include:

- Tabular visualization—time series data are presented in a table format with rows representing time periods and columns representing variables and measurements.
- Plot of measurement times—measurement times are displayed along a one-dimensional axis. Such a view helps to understand the temporal distribution of captured data.
- Linear line plot—data points are plotted and connected with straight lines, showing the data's trend and continuity over time.
- Area chart—Area charts emphasize the cumulative value or distribution over time by filling the area between the line representing the data and the x-axis.
- Bar chart—values of a specific measurement are represented as rectangular bars with corresponding height. This helps when comparing values between different time periods or categories.
- Histogram—divides the range of values into intervals and represents the distribution of data based on the count of data points falling in each interval.

Depending on the source, time series datasets can be enormous, and retrieving millions of rows from data to project the data into a chart can result in very high bandwidth between the database and the visualization system. This can be improved by aggregating the data and only visualizing datasets in the needed detail [37]. An example of such an approach is time-based grouping and aligning the time intervals with the pixel columns of the resulting chart, for which just a fraction of the detail of the whole dataset is usually needed. The value of a group can be either sampled or computed using aggregation functions, such as *min*, *max*, *avg*, *median*, *medoid* or *mode*. Ideally, the resulting chart based on aggregated data looks identical to the one based on original data and needs only a fraction of bandwidth and computing power to be displayed [37].

Tools that visualize time series data differ based on how we want to display them. Dedicated programs for data visualization are, for example, Microsoft Power BI, Microsoft Excel, or Tableau. Popular libraries used by developers include Chart.js[3] for web development, and Plotly[4] or ScottPlot[5] for desktop development.

[3] https://www.chartjs.org/
[4] https://plotly.com/
[5] https://scottplot.net/

Chapter 4

Analysis

This thesis aims to obtain, process, and visualize diagnostic data captured by the Z-Bionics Z-Arm prosthesis. The target audience can be split into two groups—patients using Z-Bionics prostheses and Z-Bionics staff using the captured data to improve those prostheses. The use cases among these groups are different.

4.1 User and Company Requirements

The proposed solution will have to follow certain requirements, both by the users—patients and staff, as well as by the Z-Bionics company. Performing analysis of some of the Z-Bionics patients determined the following:

- Patients are suffering from an upper limb amputation, both congenital and traumatic.
- The amputation extent may vary from the wrist all the way up to the shoulder.
- Patients' age ranges from 15-70 years of both sexes.
- No technical proficiency can be expected.

Patients might also be suffering from other diseases or abnormalities, which puts further requirements on the accessibility of the chosen solution. Since patients use prostheses, diagnostic data collection happens while the prosthesis is physically with them, influencing how the data are collected and sent to Z-Bionics servers. Finally, the data collected about prosthesis usage is classified as sensitive personal data of the patients and must be adequately handled.

Staff, on the other hand, is expected to have at least basic technical skills and can, if necessary, undergo additional training. They mainly use laptops and desktops, so the diagnostic data visualization should be able to use the larger screen effectively. The company's requirements include:

- Existing tools should be thoroughly analyzed and used, if possible.
- Additional server expenses should kept minimal.
- The solution that requires the least research and development resources should be chosen unless it has significant benefits compared to alternatives.
- If a mobile application is developed, it has to support both Android and iOS.

4.2 Current State

Z-Bionics uses several internal tools that aid with patient management, inventory, claims management, research, and development. Some of the tools are from 3rd parties, while others are developed internally. Those include Z-Hub, a web application currently mainly used for management purposes, and Z-Arm Configurator, a combination of client software and a website, which allows configuration and monitoring of a Z-Arm prosthesis using a computer and a cable connection via USB.

The Z-Arm prosthesis comes with a custom-made PCB with firmware that can capture diagnostic data and store them on 512kB storage. Furthermore, it is equipped with a USB C port and a Bluetooth module.

4.2.1 Z-Bionics Hub

The Z-Bionics Hub is an internal application in the form of a website that is used by Z-Bionics staff and stores records of staff members, patients, insurance claims, manufactured prostheses, and components in stock. It is based on the ASP.NET Blazor Server[1] and ASP.NET Razor[2], a C# framework. Authentication is implemented using Duende IdentityServer[3] and supports both OAuth 2.0 and OpenID Connect, while a Microsoft SQL database handles the data storage. UI is based on the Bootstrap framework and UI component kit from Syncfusion[4].

The Hub uses a three-layered architecture, individual components are pictured in Figure 4.1. Currently, the hub does not contain any used API endpoints because ASP.NET Razor, the framework used for login and user management pages, is server-side, and ASP.NET Blazor Server used for main content handles communication between browser and server via SignalR, a library for real-time communication. Therefore, neither of the UI components needs to use REST API to access the database, and if communication with the Hub will be required, suitable endpoints will need to be implemented.

[1] https://learn.microsoft.com/en-us/aspnet/core/blazor/hosting-models

[2] https://learn.microsoft.com/en-us/aspnet/core/tutorials/choose-web-ui#aspnet-core-razor-pages

[3] https://duendesoftware.com/products/identityserver

[4] https://www.syncfusion.com/blazor-components

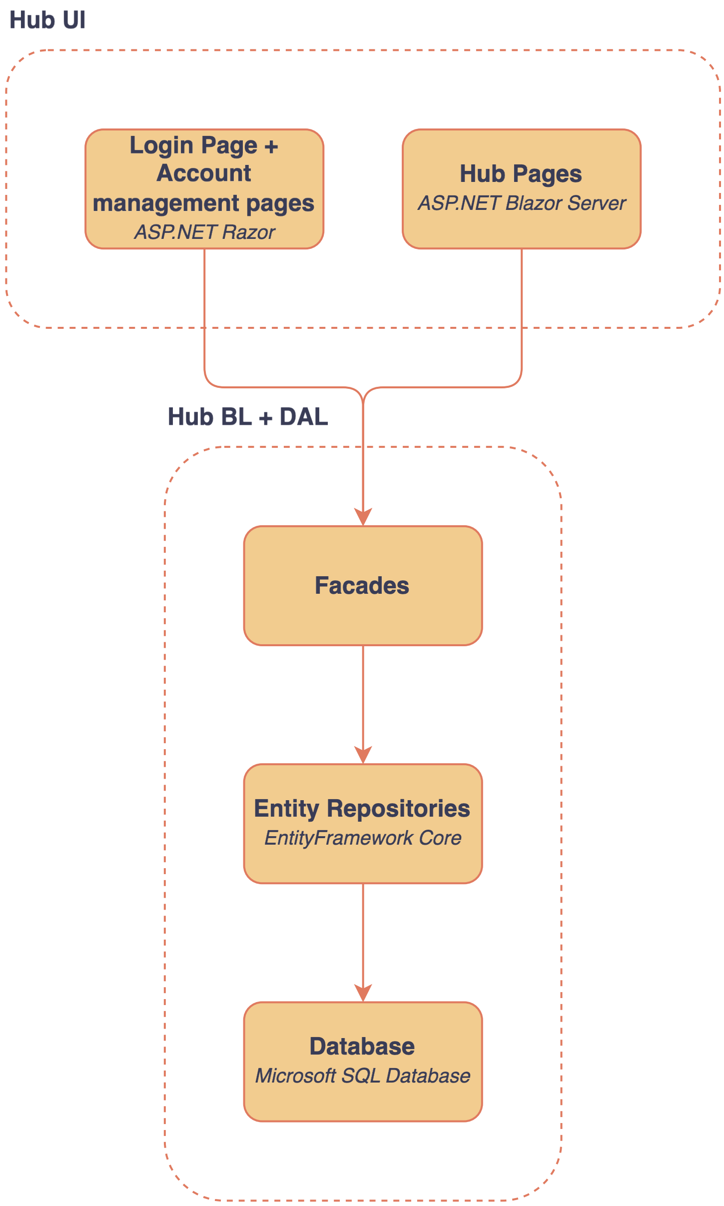

Figure 4.1: Simplified overview of current Z-Bionics Hub architecture

4.2.2 Diagnostic Data Capture

The prosthesis firmware captures and temporarily stores diagnostic data related to various prosthesis events and states. Some of the data is logged periodically, while an event triggers the collection of others. The types of diagnostic data are:

- Battery report—logged periodically every 10 minutes when the prosthesis is powered by the battery and every minute when charging. It contains information about the charging status, current charge level, battery temperature, internal resistance, and usable capacity.

- Power state report—generated when the prosthesis is turned on or off and only contains the power state.
- Motor anomaly report—logged when an abnormal motor behaviour is detected. This includes overshooting the target position, getting out of bounds, being unable to reach or overshooting the target speed. The report contains the index of the motor in question, anomaly type, position history of the part being controlled by the motor (e.g. finger), history of force applied by the motor, and the motor temperature.
- Grip mode usage report—generated every time the patient switches the prosthesis grip mode and contains basic information about the current mode.
- Snapshot report—the only type of log that is generated manually by the patient. Patients who notice a prosthesis's unexpected behaviour can easily create a snapshot by triple-pressing the main control button on the prosthesis. The report contains a short history of myo signal, motor position, force, motor temperature, and selected grip.

All reports contain a timestamp of when they were generated. After the report is generated, it is stored in CBOR[5] on the prosthesis internal storage. After the storage reaches its capacity, the oldest reports are deleted. A different retention priority can be set for various report types, thanks to which snapshots reports, for example, can be stored for considerably longer than less important battery reports.

4.2.3 Obtaining the Diagnostic Data

In order to store, process, and visualize the captured diagnostics, a reliable and unobtrusive way to obtain and upload them to Z-Bionics servers needs to be established. This can be done using peripherals already installed on prostheses, such as the USB C port and Bluetooth module, or by introducing new ones—in such cases, the feasibility of retrofitting them on existing prostheses needs to be considered.

The data has to be obtained frequently due to storage limitations of the prosthesis' hardware. The MCU has 512 kBs of storage dedicated to diagnostic data. The size of the reports ranges from 12 B up to 256 B, excluding storage overhead, which virtually doubles the size of storage needed to store a report. With an average of 624 battery reports, 360 grip mode usage reports, 180 myo status reports, 100 motor anomaly reports, 20 power state reports, and 0.5 snapshot reports every day, as determined by internal analysis, the prosthesis is only capable of storing 5-6 days worth of diagnostic data. The next generation of the prosthesis is planned to get 16-128 MBs of storage, which would be sufficient for over a year. However, since the additional storage cannot be retrofitted to existing and currently manufactured prostheses, the method of obtaining the diagnostic data must consider the need to get the data from the prosthesis every few days.

Obtaining the Diagnostic Data Using USB

The USB C connector is located on the main control board and is easily accessible via a dedicated opening in the casing. It is already used for battery charging, prosthesis configuration, debugging, and firmware updates, making it a good candidate for obtaining diagnostic data from the prosthesis. This can be achieved by connecting the prosthesis to a

[5] https://www.rfc-editor.org/rfc/rfc8949.html

computer and then using an application installed on it to communicate with the prosthesis, get the diagnostic data, and securely upload it to the Z-Bionics server. Such workflow appears to be ideal for getting the diagnostic data by the Z-Bionics staff. On the other hand, having to connect the prosthesis to a computer every few days and run an application to it to upload the diagnostic data is uncomfortable and annoying for patients.

An alternate way of uploading the diagnostic logs via USB has been proposed to address this issue. A special charging adapter, capable of connecting to the Internet via Wi-Fi, could be used. It would automatically obtain and upload the logs from the prosthesis during the charging process. This approach makes the process considerably less annoying for the patient, though it still comes with some limitations, the most notable one being the need for the charging adapter to be configured to use the patient's Wi-Fi. An intuitive way of configuring the adapter by the patient would need to be researched and implemented along with the adapter itself.

Obtaining the Diagnostic Data via IoT Network

To avoid having the patient configure and regularly use anything other than the prosthesis itself and still get the diagnostic data, an IoT network, such as Cat-M or LoRaWAN, could be utilized. It requires no action from the patient and, therefore, doesn't affect the overall comfort of prosthesis usage. On the other hand, it can only be used on new prostheses with the necessary hardware to connect to an IoT network since retrofitting is not possible due to space restraints, and the cost of such hardware and network usage fees must also be considered. Furthermore, the IoT connection likely won't be available when the patient travels abroad, which is not the case when using Bluetooth, as long as the phone is connected to the Internet.

Obtaining the Diagnostic Data via Bluetooth

Another approach that might address inconveniences related to USB transfer is Bluetooth. The prosthesis is equipped with a Bluetooth Low Energy module, which is currently not being used for any operations. If the Bluetooth functionality was implemented in the FW, an application for mobile phones could be utilized. The application would establish a connection with the prosthesis and extract the logs out of it. This approach appears to be significantly more convenient for patients than USB transfer; however, it still requires action on their part—pairing the application with the prosthesis and frequently opening it. The inconvenience of having to open the application frequently could be mitigated by also integrating the prosthesis configuration into the application.

A disadvantage of Bluetooth technology, and an issue that would have to be addressed if Bluetooth were to be chosen, is security. Since the prosthesis doesn't have a display or a keyboard, it has to accept any incoming connection request, which comes with a risk of sensitive personal data theft. Furthermore, if the prosthesis could also be configured via Bluetooth, the attacker would be able to control the motors on the prosthesis, which are strong enough to cause injury to the patient or others.

Chapter 5

Proposed Solution

Based on the findings made in Chapter 4 and a discussion with colleagues, I have decided to select a mobile application and Bluetooth as the preferred way to obtain diagnostic data from the Z-Arm prosthesis. When combined with additional features that allow easy prosthesis customization, the application appears to be the most convenient way. Unlike other approaches, such as an IoT network, it does not require any hardware changes to the prosthesis, allowing its use on already manufactured prostheses after a firmware update.

As depicted in Figure 5.1, the extension of Z-Bionics Hub will be in the form of new UI pages, a REST API, and an additional database dedicated to time series data—MongoDB. The authentication will be handled by an already implemented SSO system based on the Duende IdentityServer framework.

5.1 Mobile Application

The application is set to have additional features apart from obtaining the diagnostic data. As mentioned in Section 4.2.2, the prosthesis can capture snapshot reports when the patient encounters unexpected behaviour of the prosthesis. The application should allow the patient to include a comment describing the issue, which would greatly help the staff with finding the cause of the problem. Other additional features include:

- additional motor control (controlling wrist and elbow motors),
- ability to customize grips, save grip presets, and assign them to individual grip slots,
- myo sensor settings—sensitivity, short and long gestures,
- customization of other miscellaneous features, such as automatic standby mode and LED indicator brightness,
- low battery and charging status notifications,
- manuals.

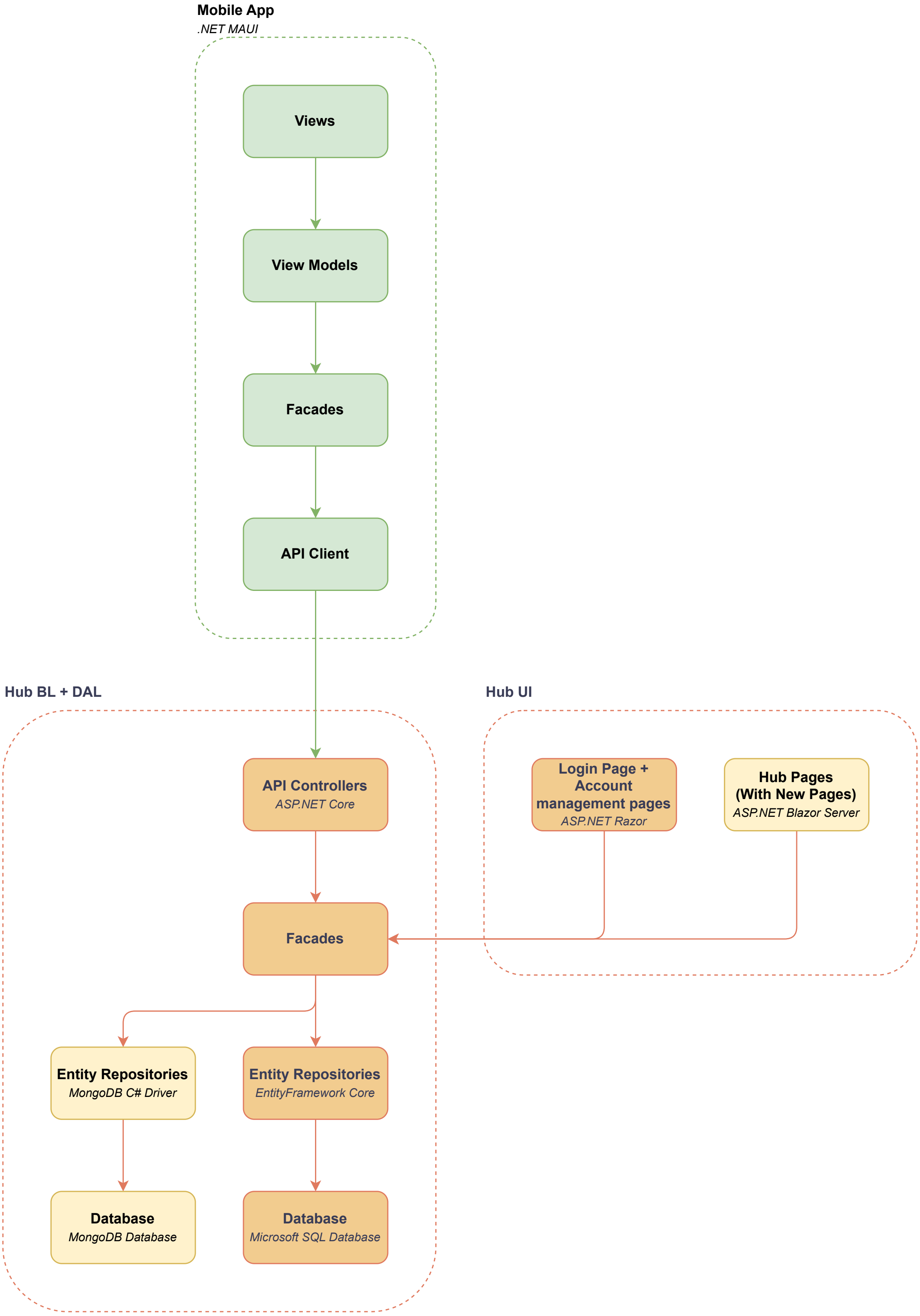

Figure 5.1: Simplified overview of the proposed changes to the Z-Bionics Hub

Based on these requirements, a Figma design has been created, and selected views can be found in Appendix A. Features were split into logical groups and given their own page. Figure A.1 shows the application's menu. Other notable pages include the homepage (figure A.2), which allows for quick grip switching as well as additional motor control; the grip configuration page (figure A.3) for more advanced grip customization, including the option to use presets; and the snapshots page A.4), which shows snapshots captured by the patient including date and time to help the patient remember why they triggered the snapshot, an option to delete the snapshot if it was created unintentionally, and finally, the option to leave a comment describing the problem the patient encountered.

5.1.1 Bluetooth Communication

The application will use Bluetooth Low Energy to communicate with the prosthesis. The communication will use CBOR[1] to transfer data more efficiently (at the cost of user readability when debugging).

Since the prosthesis has no display or keyboard, it does not do any authentication at the Bluetooth level. However, after a new device is paired, its identity is verified using a one-time password. The OTP generation is challenge-based, meaning that the prosthesis generates a number based on a secret and sends it to the mobile application, which sends it to the Z-Bionics Hub API, which generates an answer number and sends it back to the mobile application. The application then sends the answer to the prosthesis, which validates it. This approach avoids the need to share the secret anywhere apart from the prosthesis and the database the API is using, and, unlike time-based OTP, it can work even if the internal clock in the prosthesis is out of sync.

5.2 Z-Bionics Hub

The Z-Bionics Hub will be extended with appropriate API endpoints for communication with the mobile application, corresponding business logic, and data storage infrastructure to process and store obtained diagnostic data and new UI pages to display the collected diagnostic data.

5.2.1 API and Database Extension

Each diagnostic report type has a corresponding model and entity. Models are shared with the mobile application and used in rest communication, while entities will set the data format to be stored in the database. All MongoDB-related logic will be located in the Diagnostics DAL project and called from business logic facades. Mainly, CRUD operations will be implemented in the first iteration of the extended Hub, and a retention strategy will be set up for the time-series collections in the MongoDB database.

Apart from handling diagnostic data, the API will also handle the authorization and authentication of patients, both when using the provided API endpoints and when pairing with the prosthesis. The prosthesis authentication will be handled by storing a secret for the aforementioned challenge-based OTP and providing the patient's mobile application with a response generated using the provided challenge and the secret if successfully authenticated. This reduces the risk of the prosthesis being hacked and misused and also significantly reduces its usability and value in case it gets stolen from a patient.

[1] https://cbor.io/

5.2.2 UI Extension

The Hub will be extended with pages for prosthesis management. The structure will be similar to other existing management pages, visible in Figures 5.2 and 5.3, such as user management pages. They will consist of three views—a list with a filtering option, an edit page, and a create page. Edit and create pages will be made as forms, and the edit page will contain a delete button, which will show a confirmation popup before deleting the prosthesis.

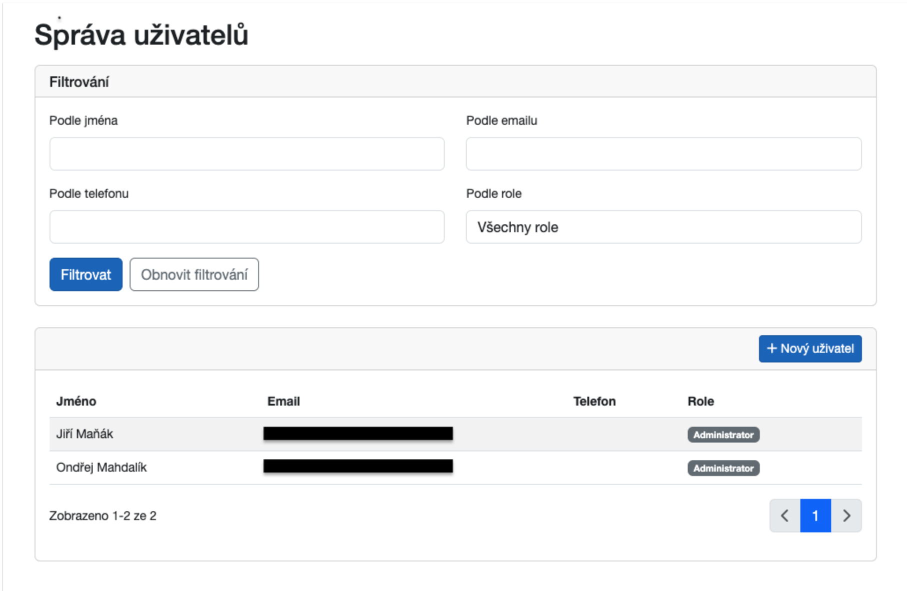

Figure 5.2: Screenshot of Z-Bionics Hub‘s user list page

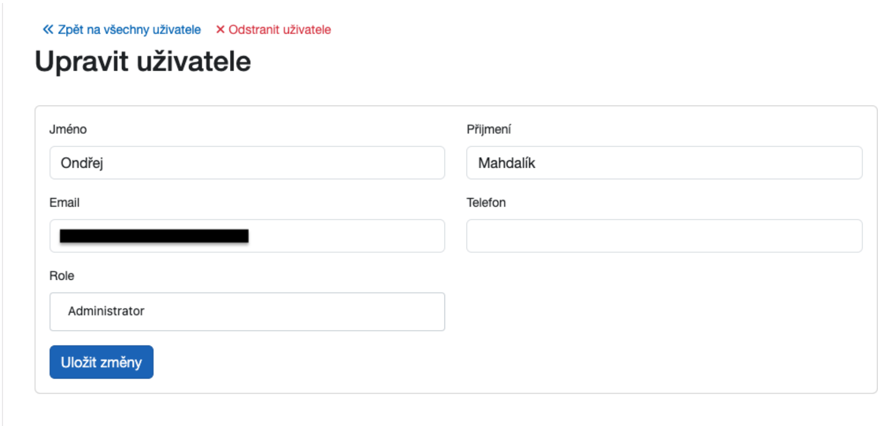

Figure 5.3: Screenshot of Z-Bionics Hub‘s user edit page

The diagnostics visualization should contain an appropriate way of selecting the desired prosthesis and a dropdown for selecting the time range of displayed data. The displayed diagnostics will be split into several categories to make navigation more efficient:

- **Summary**—a category containing general usage data, such as usage, battery and charging history.

- **Anomalies**—a category dedicated to motor anomalies.
- **Snapshots**—a category with snapshot information, similar to anomalies.
- **Other Information**—a category containing various other logged data and long-time statistics.

The summary component (Figure B.1) will contain a chart with a history of battery-related information, such as the battery level, temperature, and internal resistance. It will also highlight charging intervals. The specific way of highlighting will be dependent on the selected chart library. It will also show usage information, such as a sum of grip activations, the used grip strengths or the finger-easing feature usage.

The motor anomaly (Figure B.2) and snapshot (Figure B.3) components will both contain a timeline with selectable markers corresponding to individual motor anomalies/snapshots, a filter with appropriate selection options, and a detail section with information related to the selected motor anomaly/snapshot. The anomaly detail will contain the type of anomaly, the affected motor number, and the history of motor values captured prior to the anomaly creation. These values include the input, position, current and movement speed. The snapshot detail will contain the patient's comment (if submitted), the myo signal history, and statistical information about myo signals (minimum, average and maximum amplitudes, and reliability score). Finally, the Other Information component (Figure B.4) will contain a chart with the battery condition that will be calculated from battery reports. And plain text statistics, such as total grip usage count, power-on count, and the charging cycle count.

Chapter 6

Implementation

The implementation of the proposed changes was split into individual milestones, which would allow for better coordination with firmware developers, who made changes to the prosthesis' firmware based on my needs. The milestones were as follows:

1. Implementing mobile application front-end according to the wireframe.
2. Mocking the Bluetooth communication and adding logic to the mobile application.
3. Implementing the API and connecting it to a database.
4. Implementing Bluetooth communication into the mobile application.
5. Connecting the mobile application to the API.
6. Implementing the hub UI extension.

6.1 Mobile Application

Due to the Z-Bub using ASP.NET Blazor, a C# web framework, I decided to use .NET MAUI, which is a multi-platform mobile (and desktop) C# framework, which allows me to use a common library for shared models and significantly reduce code redundancy. The target platforms are Android and iOS. The application consists of multiple projects, representing a layered architecture. The projects are:

- **Zbionics.CustomerApp.App**—The presentation layer. Contains Views and ViewModels.
- **Zbionics.CustomerApp.BL**—The main business logic. Contains facades and their interfaces and entity mappers.
- **Zbionics.CustomerApp.BL.Mock**—Contains mocked facades that implement interfaces in the main BL project.
- **Zbionics.CustomerApp.BL.Bluetooth**—The project that handles all Bluetooth pairing and communication; Contains facades and their interfaces and services related to Bluetooth.
- **Zbionics.CustomerApp.BL.Bluetooth.Mock**—Contains mocked facades of the Bluetooth business logic.

- **Zbionics.CustomerApp.DAL**—Data access layer. Handles local storage of diagnostics.
- **Zbionics.CustomerApp.Common**—A project with enums, models and interfaces that are common for the above projects.

6.1.1 User Interface

Despite the user interface source code being shared for both platforms, the application has a slightly different appearance, respecting each platform's design guidelines to a certain extent. The application makes use of various community libraries, such as UraniumUI and CommunityToolkit, which help with improving the user experience and reducing the amount of boilerplate code (especially in ViewModels).

On the first start of the application (or when a connected prosthesis is unpaired), an initial setup is shown, which guides the user in pairing the prosthesis. The user is prompted to enter their prosthesis' serial code and a one-time activation code. After the initial setup is completed, the main application interface is shown. Despite the wireframe having a menu in the form of a collapsing sidebar, the application features a toolbar at the bottom of the screen (as visible in Figure C.1). This change increases the accessibility of the application, particularly when holding the phone only with one hand.

The grip page (Figure C.2) has undergone minor changes, especially related to presets. The preset can now be saved or loaded exclusively from the selection dialog, which can be seen in Figure C.3. The button for resetting changes has been removed and replaced with a button to open the preset selection dialog, and the save button's text has been removed and replaced with an icon in order to save space. A grip can be modified by selecting the desired index in the bottom toolbar, making changes to configuration elements, and tapping on the Save button. A confirmation snackbar is shown after the prosthesis confirms that the configuration has been saved. Currently, the prosthesis only supports finger position configuration and provides mocked data for other settings.

The myo settings page (Figure C.4) features a toolbar with sensor selection and a Save button. The sliders in the wireframe have been replaced with dropdown boxes with preset values.

Lastly, the application settings page (Figure C.5) shows basic prosthesis information, such as the serial number and a static render of an mk3 prosthesis, regardless of the actual model paired. It allows for unpairing the prosthesis, configuring the application appearance, and optionally disabling the diagnostics data sharing. All notification options are disabled, as well as the option to send the application's telemetry data. The reason notifications were not implemented is the necessity for the application to constantly run in the background and either be constantly connected to the prosthesis or periodically briefly connect. Either approach would result in increased battery consumption and, therefore, has not been implemented. The controls, however, were not completely removed because more efficient ways are being investigated, with inspiration being taken from how smartwatches maintain constant Bluetooth connection with the phone.

6.1.2 Bluetooth Communication

Bluetooth communication is built on top of an open-source library Plugin.BLE[1]. A prosthesis Bluetooth module's ID is obtained during the initial setup. It is a pseudo-GUID

[1] https://github.com/dotnet-bluetooth-le/dotnet-bluetooth-le

containing the module's MAC address. This ID is used by the Bluetooth library to connect to the prosthesis. If the connection is not successful or exceeds 15 seconds, it is repeated up to three times; after that, an error message is shown to the user. Despite opting for challenge-based OPT authentication for Bluetooth communication, the communication had to be left unsecured due to a lack of support from the prosthesis firmware. While this functionality will be implemented as soon as the firmware supports it, it cannot be done within the scope of this thesis.

After a connection is established, the application starts listening for notifications on a specific service and characteristic[2]. Dedicated entities are used for Bluetooth communication and are serialized into CBOR bytes before being sent. The communication does not work on a request-response basis. The application sends requests and listens for notifications, which may be sent by the prosthesis at any time. Since messages sent via Bluetooth may be fragmented or aggravated together, the application uses a buffer and only proceeds with processing when the received message is deemed complete.

When the prosthesis is connected, the application sends a request to obtain diagnostics. The prosthesis responds with a notification containing one or more reports and waits for acknowledgement from the application, after which it deletes the sent reports and sends another batch until all reports are sent.

When the prosthesis is not connected, or the connection is lost, a dedicated overlay appears (Figure C.6), informing the user that the application is attempting to (re)connect to the prosthesis. If all three connection attempts fail, the user can repeat them by clicking on the Connect button or head to Settings and unpair the prosthesis.

6.1.3 Local Storage and API Communication

An SQLite database serves as a local cache for the reports and is accessed using EntityFramework Core. Once the reports are obtained, the prosthesis attempts to submit them to the API. If successfully uploaded, the reports are then deleted from the local database. An HTTP client injected as a singleton is used for all API communication. Once the user registers their prosthesis using its serial code and a one-time password, the API returns a JSON Web Token, which is securely stored using MAUI's Secure Storage[3] and set as a default header for the HTTP client and used for all further communication with API.

6.2 API and Z-Bionics Hub Extension

New controllers were implemented to handle the prosthesis registration and diagnostic data submissions. Endpoints for prosthesis registration support anonymous access, while all diagnostic endpoints require authentication. When a registration request with a prosthesis serial number and one-time activation code is received, the values are validated, and a JWT is generated. It contains a claim with the prosthesis' serial number, which can then be used when processing submitted diagnostics to link them with the correct prosthesis.

Once diagnostics are received, they are stored in a MongoDB database in appropriate collections based on the report type. These collections are configured as time series. Each report contains a timestamp, serial number of the related prosthesis and type-specific data.

[2] More information about Bluetooth services and characteristics can be found at https://www.bluetooth.com/blog/a-developers-guide-to-bluetooth/

[3] https://learn.microsoft.com/en-us/dotnet/maui/platform-integration/storage/secure-storage

The existing Microsoft SQL Server database schema was extended with the Prosthesis entity, which holds information about the prosthesis' model, serial number, Bluetooth module ID described in Section 6.1.2, patient entity's ID and other information.

6.2.1 UI Extension and Diagnostics Visualization

Pages for prosthesis management were added. They consist of a list page with a filter (Figure D.1), a new prosthesis page, and an edit page (Figure D.2). The create and edit pages feature a filterable dropdown from Syncfusion[4]. The edit page also allows for prosthesis deletion. When clicked, a confirmation dialog appears, and if confirmed, the prosthesis and all diagnostic data related to it are deleted. The diagnostic data are deleted in order to comply with GDPR, which enforces that personal data should not be kept any longer than needed.

Data visualization is handled by the Diagnostics page. It has a navbar (Figure D.3) consisting of a prosthesis dropdown that supports filtering. The prosthesis can be filtered based on its model, serial number, patient name, or Bluetooth module ID. Next to the prosthesis dropdown is a time selector, which controls how old diagnostics should be displayed. Currently, it has an option for 24 hours, seven days and 30 days. A year option may be added at a later point if deemed necessary. The dropdown is followed by navigation links to individual components—Summary, Anomalies, Snapshots and Other Information.

The summary component comprises a battery chart and multiple grip usage charts. The battery chart (Figure D.4) is a line chart with the x-axis serving as a time axis and with multiple y-axes in order to display various types of data. The data visualized in the battery chart are the state of charge, the battery temperature, and internal resistance. Furthermore, the state of the charge line changes colour depending on the charging status—orange when on battery and green when charging. When hovered, the chart displays precise values displayed in a tooltip. Usage charts (Figure D.5) consist of a pie chart displaying what grip strength is used by the patient, a column chart containing the sum of grip activation in individual time intervals, and a pie chart for finger-easing feature usage.

Anomaly (Figure D.6) and snapshot (Figure D.7) components contain a timeline in the form of a chart with markers representing individual anomalies/snapshots. A filter is also available. The anomaly component detail contains the type of anomaly, the affected motor number, and a line chart with a history of logged motor values prior to the anomaly being detected. The x-axis does not have a time unit because the final sample rate might differ based on the prosthesis model. The snapshot report details the patient's comment, a line chart with myo signal history for each sensor, and a combined chart with myo signal statistics, such as minimum, average, and maximum amplitude and reliability score computed by the prosthesis.

The other information component currently only has long-term statistics, such as the total number of power cycles and grip activations. The battery condition chart had to be left out because of a lack of data. While the prosthesis is reporting internal resistance and total battery cycles can be calculated from the battery reports, estimating the battery condition is a complex task, and every battery model behaves differently; therefore, a simple equation with currently available information would be too inaccurate. Additional research about battery behaviour and possibly more detailed logging will be needed to implement this feature.

[4] https://www.syncfusion.com/

Chapter 7

Testing

Different approaches were chosen for various layers. Backend layers, such as the Z-Bionics Hub's BL, DAL and API, were suitable for unit and end-to-end tests, while Bluetooth communication and the front end of both mobile applications and Hub's extensions were tested manually. The front end needed to be tested for any present bugs and to ensure that the user experience was optimal, especially regarding accessibility.

7.1 Unit and End-to-end Testing

Z-Bionics Hub's CRUD operations were tested using testing library xUnit[1]. The automated testing was focused on prosthesis entities and diagnostic reports, with a total of over 40 unit tests focused on the following scenarios:

- getting an existing item,
- attempting to get a non-existing item,
- adding a new item,
- adding multiple items,
- modifying an existing item,
- deleting an existing item,
- attempting to delete a non-existing item.

The API endpoints were covered by E2E tests using both unauthorized and authorized HTTP clients. Various scenarios were tested, including attempts to submit diagnostic reports with falsified prosthesis numbers or invalid JWT. Unfortunately, the mobile application's DAL had to be manually tested because running Unit tests on a mobile phone proved more tricky than initially expected.

7.2 Mobile Application Testing

The mobile application has been tested manually by the developer and other Z-Bionics employees and it is also currently being tested by one patient, with more patients to come

[1] https://xunit.net/

in the near future. The main areas of focus are the user interface, Bluetooth communication, local storage and API communication.

7.2.1 Bluetooth Communication Testing

Bluetooth communication can often be inconsistent and contains many elements that can behave unexpectedly. Tested scenarios include simulating loss of connection by unplugging the prosthesis from the battery and checking if the application manages to reconnect, sending multiple messages in quick succession and checking if the prosthesis receives them and vice-versa.

While the connection stability has been based on the testing results greatly improved compared to the initial version, a number of bugs still persist. Most of the bugs don't seem to be related to the application but to the prosthesis' Bluetooth module or its firmware. For example, the prosthesis can sometimes stop sending notifications even when supposed to (e.g. when a power state or grip index is changed, either via a Bluetooth request or directly on the prosthesis by pressing a button). These prosthesis-related bugs usually disappear when the prosthesis is reset by disconnecting the battery. However, during one testing session, a prosthesis' Bluetooth module completely stopped accepting incoming connections despite multiple resets and firmware rewrites. This behaviour could not be replicated during any other testing session, which leaves the bug potentially unresolved.

7.2.2 User Interface Testing With a Patient

One patient who is regularly using their prosthesis has been selected to participate in mobile application testing. At the time of writing this thesis, the testing has not been yet concluded, so no definitive feedback has been obtained. However, the patient was happy to share their first impressions after a few days of using the application, and their most notable comments were:

- The Bluetooth connection time can sometimes take over 30 seconds.
- The user interface is pretty accessible when holding the phone in one hand. The presence of most buttons in the bottom half and a minimal amount of buttons near the edges of the phone makes handling more comfortable compared to some other applications used by the client.
- The grip configuration is intuitive; however, a preview of the target fingers' positions would be very welcome.
- The application sometimes doesn't show the connecting overlay when a connection is lost or shows it late.
- Battery life of the phone is slightly reduced (more testing is needed to judge if it is caused by the application itself or just increased usage).

7.3 Z-Bionics Hub Extension UI Testing

With the API and Business Logic code being covered by unit and E2E tests, the last part to test was the user interface. Automated UI tests using Selenium framework[2] were considered.

[2] https://www.selenium.dev/

However, due to the small number of pages and interactions on them, it was deemed not worth the effort and time needed to implement, and manual testing was chosen instead. Basic CRUD operations for prostheses were tested on respective pages, and visualizations of both seeded and real data were checked. Other Z-Bionics employees also tried the Hub extension and provided ideas for small improvements in order to improve the readability of some charts. The optimization of the visualization has not been measured; however, statistics from the hosting provider (Figure 7.1) do not suggest any significant optimization issues.

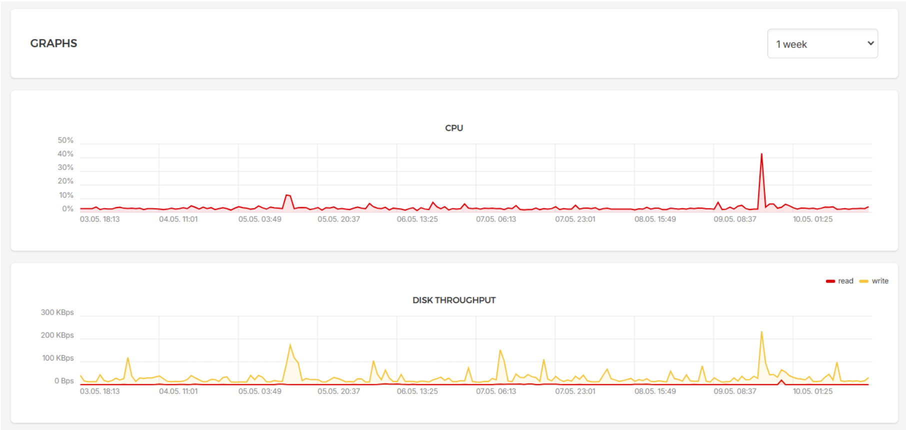

Figure 7.1: Resource usage from hosting provider's dashboard

Chapter 8

Conclusion

The aim of this thesis was to design and implement a way of obtaining, processing and visualizing diagnostic data from a bionic prosthesis. Research into bionic prostheses, sensitive healthcare data handling and time series was conducted in order to get a sufficient understanding of the problem. Patient and staff needs were analysed, various ways of obtaining diagnostic data from the prosthesis were examined and compared, and relevant existing tools used by Z-Bionics were investigated.

A solution consisting of a mobile application utilizing Bluetooth to communicate with the bionic prosthesis and a Z-Bionics Hub extension was proposed and implemented. The application turned out to be the most crucial part, thanks to additional features, such as the prosthesis configuration, that significantly improve the comfort and intuitiveness of the prosthesis usage. The captured diagnostics, on the other hand, are expected to help with the long-term satisfaction of patients by allowing Z-Bionics employees to proactively assist patients when improper usage or a defect is detected before these inconveniences result in the patient putting the prosthesis away. User-submitted snapshots are especially helpful because they allow the patient to describe their issue and make them feel more involved in the troubleshooting process.

The objectives set for this thesis can be, therefore, considered met. The results were also presented at the Excel@FIT conference [46]. The development process, however, doesn't end with the publication of this thesis. A lot of work is yet to be put into polishing already implemented features, improving the stability of the solution and making it ready for use by more patients. The Z-Bionics Hub currently barely processes the obtained diagnostics. Much more information could be potentially extracted from the data than currently is, and more advanced techniques, such as machine learning, could be utilized. Yet, even in the current state, all implemented parts can be used on a daily basis by both employees and patients, who can benefit from the value brought by the implemented solution.

Bibliography

[1] Abanda, A., Mori, U. and Lozano, J. A. *A review on distance based time series classification*. 2018. [Online, visited 12/16/2023].

[2] Abrams, Z. A New Era for Bionic Limbs. march 2023. [Online, visited 11/06/2023]. Available at: https://www.embs.org/pulse/articles/a-new-era-for-bionic-limbs/.

[3] Alessia, P. Time series with MongoDB. *Data Reply IT | DataTech*. october 2023. [Online, visited 12/28/2023]. Available at: https://medium.com/data-reply-it-datatech/time-series-with-mongodb-dd60f8d6acd6.

[4] Amidon, A. *A Brief Survey of Time Series Classification Algorithms*. September 2020. [Online, visited 10/07/2023]. Available at: https://towardsdatascience.com/a-brief-introduction-to-time-series-classification-algorithms-7b4284d31b97.

[5] Anadiotis, G. *Processing time series data: What are the options?* September 2018. [Online, visited 10/04/2023]. Available at: https://www.zdnet.com/article/processing-time-series-data-what-are-the-options.

[6] Analytics, E. *Everything You Need to Know About Time Series Analysis – A Primer*. 2021. [Online, visited 09/18/2023]. Available at: https://www.expressanalytics.com/blog/time-series-analysis/.

[7] Asyali, M. H., Yilmaz, M., Tokmakci, M., Sedef, K., Aksebzeci, B. H. et al. Design and implementation of a voice-controlled prosthetic hand. *Turkish Journal of Electrical Engineering and Computer Sciences*. 2011, vol. 19, no. 1, p. 33–46.

[8] Bio-Medical Instruments. *Myoscan sEMG Sensor for FlexComp/Myotrac*. 2023. [Online, visited 12/02/2023]. Available at: https://bio-medical.com/myoscan-semg-sensor-for-flexcompmyotrac.html.

[9] Biometrics Ltd. *Surface EMG Sensors*. 2023. [Online, visited 12/02/2023]. Available at: https://www.biometricsltd.com/surface-emg-sensor.htm.

[10] Bradshaw, S., Brazil, E. and Chodorow, K. *MongoDB: the definitive guide: powerful and scalable data storage*. O'Reilly Media, 2019.

[11] Calado, A., Soares, F. and Matos, D. A Review on Commercially Available Anthropomorphic Myoelectric Prosthetic Hands, Pattern-Recognition-Based Microcontrollers and sEMG Sensors used for Prosthetic Control. In: *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. 2019, p. 1–6. DOI: 10.1109/ICARSC.2019.8733629.

[12] Ccorimanya, L., Watanabe, R., Hassan, M., Hada, Y. and Suzuki, K. Design of a myoelectric 3D-printed prosthesis for a child with upper limb congenital amputation. In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2019, p. 5394–5398. DOI: 10.1109/EMBC.2019.8857899.

[13] De Luca, C. J. The use of surface electromyography in biomechanics. *Journal of applied biomechanics*. Human Kinetics, Inc. 1997, vol. 13, no. 2, p. 135–163.

[14] Domingo Ferrer, J., Farràs, O., Ribes González, J. and Sánchez, D. Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges. *Computer Communications*. 2019, 140-141, p. 38–60. DOI: https://doi.org/10.1016/j.comcom.2019.04.011. ISSN 0140-3664. Available at: https://www.sciencedirect.com/science/article/pii/S0140366418310740.

[15] ELEMYO. *Myoelectric sensor MYO v1.4*. 2023. [Online, visited 08/11/2023]. Available at: https://www.elemyo.com/products/myo_v1_4.

[16] Esposito, D., Andreozzi, E., Fratini, A., Gargiulo, G. D., Savino, S. et al. *A Piezoresistive Sensor to Measure Muscle Contraction and Mechanomyography*. 2018. DOI: 10.3390/s18082553. ISSN 1424-8220.

[17] Eva, G., Liese, G., Stephanie, B., Petr, H., Leslie, M. et al. Position paper on management of personal data in environment and health research in Europe. *Environment International*. 2022, vol. 165, p. 107334. DOI: https://doi.org/10.1016/j.envint.2022.107334. ISSN 0160-4120. Available at: https://www.sciencedirect.com/science/article/pii/S0160412022002616.

[18] Fang, Y., Xu, H. and Jiang, J. A Survey of Time Series Data Visualization Research. *IOP Conference Series: Materials Science and Engineering*. IOP Publishing. mar 2020, vol. 782, no. 2, p. 022013. DOI: 10.1088/1757-899X/782/2/022013. Available at: https://dx.doi.org/10.1088/1757-899X/782/2/022013.

[19] Ferigo, D., Merhi, L.-K., Pousett, B., Xiao, Z. G. and Menon, C. A Case Study of a Force-myography Controlled Bionic Hand Mitigating Limb Position Effect. *Journal of Bionic Engineering*. 2017, vol. 14, no. 4, p. 692–705. DOI: https://doi.org/10.1016/S1672-6529(16)60435-3. ISSN 1672-6529. Available at: https://www.sciencedirect.com/science/article/pii/S1672652916604353.

[20] Fundamental Rights, E. U. A. for and Europe, C. of. *Handbook on European data protection law – 2018 edition*. Publications Office of the European Union, 2018.

[21] Gangadhar, S. et al. The real time environmental time series data analysis using Influx DB. *International Journal of Advanced Scientific Innovation*. 2020, vol. 1, no. 1.

[22] Geethanjali, P. Myoelectric control of prosthetic hands: state-of-the-art review. *Medical devices (Auckland, N.Z.)*. New Zealand: [b.n.]. 2016, vol. 9, p. 247–55. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4968852/.

[23] Gilbert, S. and Lynch, N. Perspectives on the CAP Theorem. *Computer*. 2012, vol. 45, no. 2, p. 30–36. DOI: 10.1109/MC.2011.389.

[24] Gohel, V. and Mehendale, N. Review on electromyography signal acquisition and processing. *Biophysical Reviews*. 2020, vol. 12, no. 6, p. 1361–1367. DOI: 10.1007/s12551-020-00770-w. ISSN 1867-2469. Available at: https://doi.org/10.1007/s12551-020-00770-w.

[25] González Liaño, I. and Vázquez Fernández, M. Use of TimescaleDB as a database for ocean-meteorological data storage. *Instrumentation Viewpoint*. SARTI. 2021, no. 21, p. 40–41.

[26] Guglani, G. *Time Series Data Visualization: Types, Techniques & Platforms*. September 2023. [Online, visited 12/29/2023]. Available at: https://www.knowledgehut.com/blog/business-intelligence-and-visualization/time-series-data-visualization.

[27] Han, J., E, H., Le, G. and Du, J. Survey on NoSQL database. In: *2011 6th International Conference on Pervasive Computing and Applications*. 2011, p. 363–366. DOI: 10.1109/ICPCA.2011.6106531.

[28] Hebert, J. S., Elzinga, K., Chan, K. M., Olson, J. and Morhart, M. Updates in Targeted Sensory Reinnervation for Upper Limb Amputation. *Current Surgery Reports*. 2014, vol. 2, no. 3, p. 45. DOI: 10.1007/s40137-013-0045-7. ISSN 2167-4817. Available at: https://doi.org/10.1007/s40137-013-0045-7.

[29] InfluxData. *Data retention in InfluxDB*. 2023. [Online, visited 12/25/2023]. Available at: https://docs.influxdata.com/influxdb/v2/reference/internals/data-retention/.

[30] InfluxData. *Get started with InfluxDB tasks*. 2023. [Online, visited 12/25/2023]. Available at: https://docs.influxdata.com/influxdb/v2/process-data/get-started/.

[31] InfluxData. *InfluxDB data elements*. 2023. [Online, visited 12/25/2023]. Available at: https://docs.influxdata.com/influxdb/cloud/reference/key-concepts/data-elements/#sample-data.

[32] InfluxData. *InfluxDB schema design*. 2023. [Online, visited 12/25/2023]. Available at: https://docs.influxdata.com/influxdb/cloud/write-data/best-practices/schema-design/#use-tags-to-improve-query-performance.

[33] InfluxData. *Use InfluxDB client libraries*. 2023. [Online, visited 12/25/2023]. Available at: https://docs.influxdata.com/influxdb/v2/api-guide/client-libraries/.

[34] InfluxDB. *What is time series data?* 2023. [Online, visited 09/03/2023]. Available at: https://www.influxdata.com/what-is-time-series-data/.

[35] intersoft consulting. *GDPR - Right to be Informed*. 2023. [Online, visited 12/29/2023]. Available at: https://gdpr-info.eu/issues/right-to-be-informed/.

[36] JIN, H., LUO, Y., LI, P. and MATHEW, J. A Review of Secure and Privacy-Preserving Medical Data Sharing. *IEEE Access*. 2019, vol. 7, p. 61656–61669. DOI: 10.1109/ACCESS.2019.2916503.

[37] JUGEL, U., JERZAK, Z., HACKENBROICH, G. and MARKL, V. M4: a visualization-oriented time series data aggregation. *Proc. VLDB Endow.* VLDB Endowment. jun 2014, vol. 7, no. 10, p. 797–808. DOI: 10.14778/2732951.2732953. ISSN 2150-8097. Available at: https://doi.org/10.14778/2732951.2732953.

[38] KEZMANN, J. M. *All 8 Types of Time Series Classification Methods*. 2022. [Online; accessed 18-September-2023]. Available at: https://medium.com/mlearning-ai/all-8-types-of-time-series-classification-methods-2c8e4b323ea2.

[39] KUIKEN, T. A., DUMANIAN, G. A., LIPSCHUTZ, R. D., MILLER, L. A. and STUBBLEFIELD, K. The use of targeted muscle reinnervation for improved myoelectric prosthesis control in a bilateral shoulder disarticulation amputee. *Prosthetics and orthotics international*. Taylor & Francis. 2004, vol. 28, no. 3, p. 245–253.

[40] KYBERD, P. J. and POULTON, A. Use of Accelerometers in the Control of Practical Prosthetic Arms. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2017, vol. 25, no. 10, p. 1884–1891. DOI: 10.1109/TNSRE.2017.2693683.

[41] LEE, C. H. and YOON, H.-J. Medical big data: promise and challenges. *Kidney research and clinical practice*. Korea (South): [b.n.]. Mar 2017, vol. 36, p. 3–11.

[42] LEI, Y. and WU, Z. Time series classification based on statistical features. *EURASIP Journal on Wireless Communications and Networking*. 2020, vol. 2020, no. 1, p. 46. DOI: 10.1186/s13638-020-1661-4. ISSN 1687-1499. Available at: https://doi.org/10.1186/s13638-020-1661-4.

[43] LI, K., ZHANG, J., WANG, L., ZHANG, M., LI, J. et al. A review of the key technologies for sEMG-based human-robot interaction systems. *Biomedical Signal Processing and Control*. 2020, vol. 62, p. 102074. DOI: https://doi.org/10.1016/j.bspc.2020.102074. ISSN 1746-8094. Available at: https://www.sciencedirect.com/science/article/pii/S1746809420302305.

[44] LI, Q., SONG, Y. and HOU, Z. Estimation of lower limb periodic motions from sEMG using least squares support vector regression. *Neural Processing Letters*. Springer. 2015, vol. 41, no. 3, p. 371–388.

[45] LOCKERMAN, J. and KULKARNI, A. *Time-Series Compression Algorithms, Explained*. April 2020. [Online, visited 10/04/2023]. Available at: https://www.timescale.com/blog/time-series-compression-algorithms-explained.

[46] MAHDALÍK, O. Processing and Visualization of Diagnostic Data from a Bionic Upper Limb Prosthesis. In: *Excel@FIT*. 2024. Available at: https://excel.fit.vutbr.cz/2024/sbornik/.

[47] MARASCO, P. D., KIM, K., COLGATE, J. E., PESHKIN, M. A. and KUIKEN, T. A. Robotic touch shifts perception of embodiment to a prosthesis in targeted reinnervation amputees. *Brain*. january 2011, vol. 134, no. 3, p. 747–758. DOI:

10.1093/brain/awq361. ISSN 0006-8950. Available at: https://doi.org/10.1093/brain/awq361.

[48] Marinelli, A., Boccardo, N., Tessari, F., Domenico, D. D., Caserta, G. et al. Active upper limb prostheses: a review on current state and upcoming breakthroughs. *Progress in Biomedical Engineering.* IOP Publishing. jan 2023, vol. 5, no. 1, p. 012001. DOI: 10.1088/2516-1091/acac57. Available at: https://dx.doi.org/10.1088/2516-1091/acac57.

[49] Marshall, C. J. MODERN ARTIFICIAL LIMBS. THE WORK OF THE ARM-TRAINING CENTRE AT ROEHAMPTON. *The Lancet.* Elsevier. 1921, vol. 197, no. 5104, p. 1374–1376.

[50] Meier, A. and Kaufmann, M. *SQL & NoSQL databases.* Springer, 2019.

[51] MongoDB. *Time Series.* 2023. [Online, visited 12/28/2023]. Available at: https://www.mongodb.com/docs/manual/core/timeseries-collections/.

[52] Naqvi, S. N. Z., Yfantidou, S. and Zimányi, E. Time series databases and influxdb. *Studienarbeit, Université Libre de Bruxelles.* 2017, vol. 12.

[53] Nayak, A., Poriya, A. and Poojary, D. Type of NOSQL databases and its comparison with relational databases. *International Journal of Applied Information Systems.* 2013, vol. 5, no. 4, p. 16–19.

[54] Nazari, V. and Zheng, Y.-P. *Controlling Upper Limb Prostheses Using Sonomyography (SMG): A Review.* 2023. DOI: 10.3390/s23041885. ISSN 1424-8220.

[55] Olman, C. *TARGETED SENSORY RE-INNERVATION.* 2023. [Online, visited 12/02/2023]. Available at: https://pressbooks.umn.edu/sensationandperception/chapter/targeted-sensory-reinnervation-draft/.

[56] Oracle. *What is a Relational Database (RDBMS)?* 2023. [Online, visited 10/08/2023]. Available at: www.oracle.com/database/what-is-a-relational-database.

[57] Ortiz Catalan, M., Zbinden, J., Millenaar, J., D'Accolti, D., Controzzi, M. et al. A highly integrated bionic hand with neural control and feedback for use in daily life. *Science Robotics.* 2023, vol. 8, no. 83, p. eadf7360. DOI: 10.1126/scirobotics.adf7360. Available at: https://www.science.org/doi/abs/10.1126/scirobotics.adf7360.

[58] OurPCB Tech. Ltd.. *EMG Sensors: How Do They Work?* 2023. [Online, visited 11/20/2023]. Available at: https://www.ourpcb.com/emg-sensors.html.

[59] Parsons, K., McCormac, A., Butavicius, M. and Ferguson, L. *Human Factors and Information Security: Individual,Culture and Security Environment.* techreport ADA535944. Defence Science and Technology Organisation, Command, Control, Communications and Intelligence Division, october 2010. [Online, visited 12/29/2023]. Available at: https://apps.dtic.mil/sti/citations/ADA535944.

[60] Pawar, A., Ahirrao, S. and Churi, P. P. Anonymization Techniques for Protecting Privacy: A Survey. In: *2018 IEEE Punecon.* 2018, p. 1–6. DOI: 10.1109/PUNECON.2018.8745425.

[61] Phaltankar, A., Ahsan, J., Harrison, M. and Nedov, L. *MongoDB Fundamentals: A hands-on guide to using MongoDB and Atlas in the real world.* Packt Publishing Ltd, 2020.

[62] Płuciennik, E. and Zgorzałek, K. The Multi-model Databases – A Review. In: Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak Mrozek, B. and Kostrzewa, D., ed. *Beyond Databases, Architectures and Structures. Towards Efficient Solutions for Data Analysis and Knowledge Representation.* Cham: Springer International Publishing, 2017, p. 141–152. ISBN 978-3-319-58274-0.

[63] Putti, V. Historical prostheses. *Journal of Hand Surgery.* SAGE Publications Sage UK: London, England. 2005, vol. 30, no. 3, p. 310–325.

[64] Salminger, S., Stino, H., Pichler, L. H., Gstoettner, C., Sturma, A. et al. Current rates of prosthetic usage in upper-limb amputees – have innovations had an impact on device acceptance? *Disability and Rehabilitation.* Taylor & Francis. 2022, vol. 44, no. 14, p. 3708–3713. DOI: 10.1080/09638288.2020.1866684. PMID: 33377803. Available at: https://doi.org/10.1080/09638288.2020.1866684.

[65] Scott, R. Myoelectric control of prostheses: A brief history. In: Myoelectric Symposium. 1992.

[66] Shahsavari, H., Matourypour, P., Ghiyasvandian, S., Ghorbani, A., Bakhshi, F. et al. Upper limb amputation; Care needs for reintegration to life: An integrative review. *International Journal of Orthopaedic and Trauma Nursing.* 2020, vol. 38, p. 100773. DOI: https://doi.org/10.1016/j.ijotn.2020.100773. ISSN 1878-1241. Available at: https://www.sciencedirect.com/science/article/pii/S1878124119301911.

[67] Sharma, A., Singh, A., Sharma, N., Kaushik, I. and Bhushan, B. Security Countermeasures in Web Based Application. In: *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT).* 2019, vol. 1, p. 1236–1241. DOI: 10.1109/ICICICT46008.2019.8993141.

[68] Shen, L., Lou, Y., Chen, Y., Lu, M. and Ye, F. Meteorological Sensor Data Storage Mechanism Based on TimescaleDB and Kafka. In: Cheng, X., Jing, W., Song, X. and Lu, Z., ed. *Data Science.* Singapore: Springer Singapore, 2019, p. 137–147. ISBN 978-981-15-0118-0.

[69] Sherman, E. D. A Russian bioelectric-controlled prosthesis: Report of a research team from the Rehabilitation Institute of Montreal. *Canadian Medical Association Journal.* Canadian Medical Association. 1964, vol. 91, no. 24, p. 1268.

[70] Statistics and Data. *The Most Popular Databases – 2006/2023.* 2023. [Online, visited 12/25/2023]. Available at: https://statisticsanddata.org/data/the-most-popular-databases-2006-2023/.

[71] Stögner, V. A., Megerle, K., Krezdorn, N. and Vogt, P. M. Major traumatic amputations and replantations of the upper extremity in Germany - National quality reports 2014-2018. *JPRAS open.* Netherlands: [b.n.]. Jun 2022, vol. 32, p. 98–110. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8956841/.

[72] Terrazas Rodas, D. and Lopez Chavez, L. Techniques and Control Systems in Upper-Limb Bionic Prostheses: A Technological Review. In: *2022 International Conference on Smart Systems and Power Management (IC2SPM)*. Nov 2022, p. 110–116. DOI: 10.1109/IC2SPM56638.2022.9988849.

[73] Timescale. *About hypertables*. 2023. [Online, visited 12/28/2023]. Available at: https://docs.timescale.com/use-timescale/latest/hypertables/about-hypertables/.

[74] Toth, A. *Time-Series Database: An Explainer*. February 2023. [Online, visited 10/08/2023]. Available at: https://www.timescale.com/blog/what-is-a-time-series-database/.

[75] U.S. Government Accountability Office. *SolarWinds Cyberattack Demands Significant Federal and Private-Sector Response (infographic)*. April 2021. [Online, visited 05/13/2024]. Available at: https://www.gao.gov/blog/solarwinds-cyberattack-demands-significant-federal-and-private-sector-response-infographic.

[76] Vu, P. P., Chestek, C. A., Nason, S. R., Kung, T. A., Kemp, S. W. et al. The future of upper extremity rehabilitation robotics: research and practice. *Muscle & Nerve*. 2020, vol. 61, no. 6, p. 708–718. DOI: https://doi.org/10.1002/mus.26860. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/mus.26860.

[77] Wikipedia contributors. *Electromyography — Wikipedia, The Free Encyclopedia*. 2023. [Online, visited 11/20/2023]. Available at: https://en.wikipedia.org/w/index.php?title=Electromyography&oldid=1182725488.

[78] Wikipedia contributors. *Relational database — Wikipedia, The Free Encyclopedia*. 2023. [Online, visited 10/07/2023]. Available at: https://en.wikipedia.org/w/index.php?title=Relational_database&oldid=1177435717.

[79] Wikipedia contributors. *Time series — Wikipedia, The Free Encyclopedia*. 2023. [Online, visited 09/03/2023]. Available at: https://en.wikipedia.org/w/index.php?title=Time_series&oldid=1172322375.

[80] Williams, H. E., Shehata, A. W., Dawson, M. R., Scheme, E., Hebert, J. S. et al. Recurrent Convolutional Neural Networks as an Approach to Position-Aware Myoelectric Prosthesis Control. *IEEE Transactions on Biomedical Engineering*. 2022, vol. 69, no. 7, p. 2243–2255. DOI: 10.1109/TBME.2022.3140269.

[81] Xing, Z., Pei, J. and Keogh, E. A Brief Survey on Sequence Classification. *SIGKDD Explor. Newsl.* New York, NY, USA: Association for Computing Machinery. nov 2010, vol. 12, no. 1, p. 40–48. DOI: 10.1145/1882471.1882478. ISSN 1931-0145. Available at: https://doi.org/10.1145/1882471.1882478.

[82] Z-Bionics s.r.o.. *Internal Z-Bionics resources*. 2023.

[83] Zuo, K. J. and Olson, J. L. The evolution of functional hand replacement: From iron prostheses to hand transplantation. *Plastic surgery (Oakville, Ont.)*. United States: [b.n.]. Spring 2014, vol. 22, p. 44–51.

Appendices

List of Appendices

Appendix A

Mobile Application Wireframe

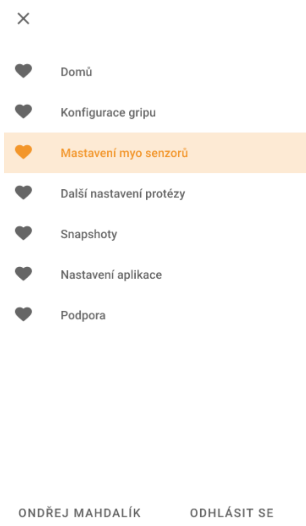

Figure A.1: Menu of the mobile application

Figure A.2: The mobile application homepage featuring additional motor control and grip selection

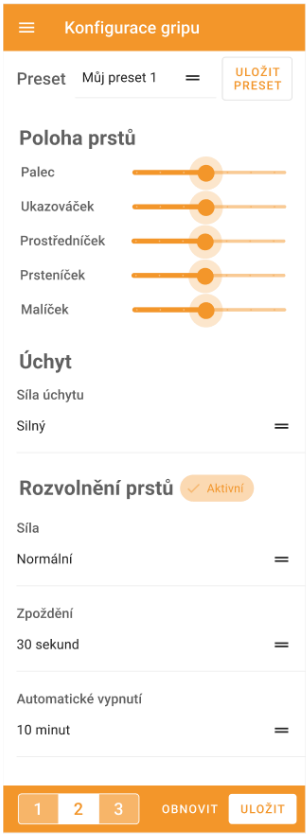

Figure A.3: The grip configuration page in the mobile application which allows for grip customization, including creating presets

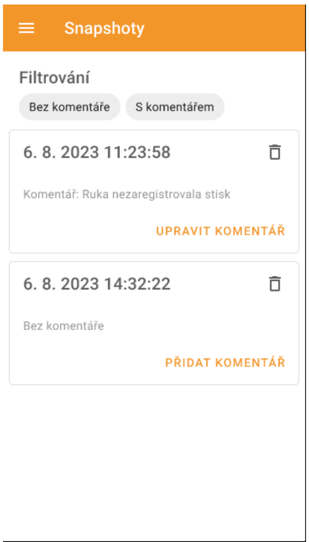

Figure A.4: The mobile application snapshot page for managing created snapshots and optionally adding comments for issue description

Appendix B

Z-Bionics Hub Extension Wireframe

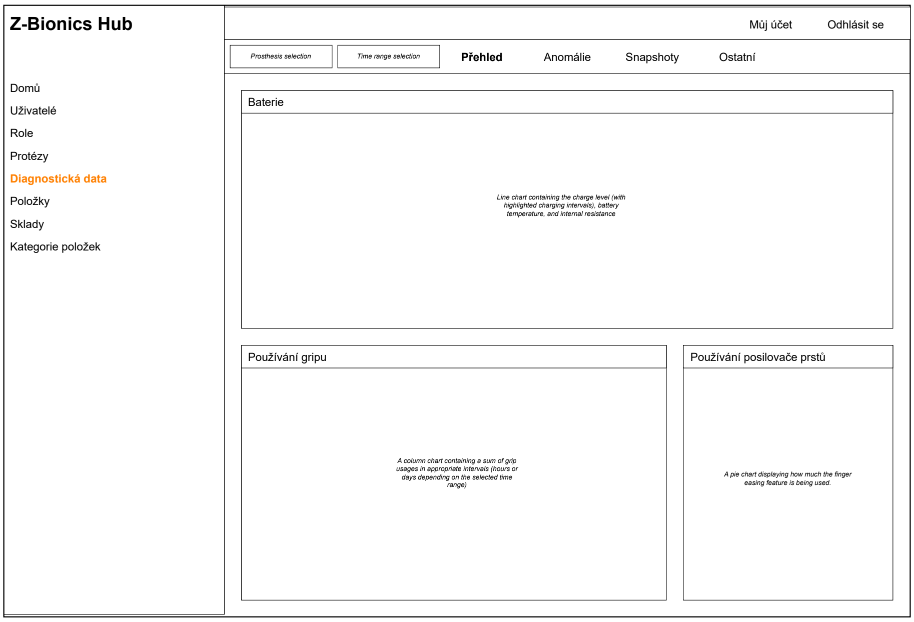

Figure B.1: Wireframe of diagnostics summary page

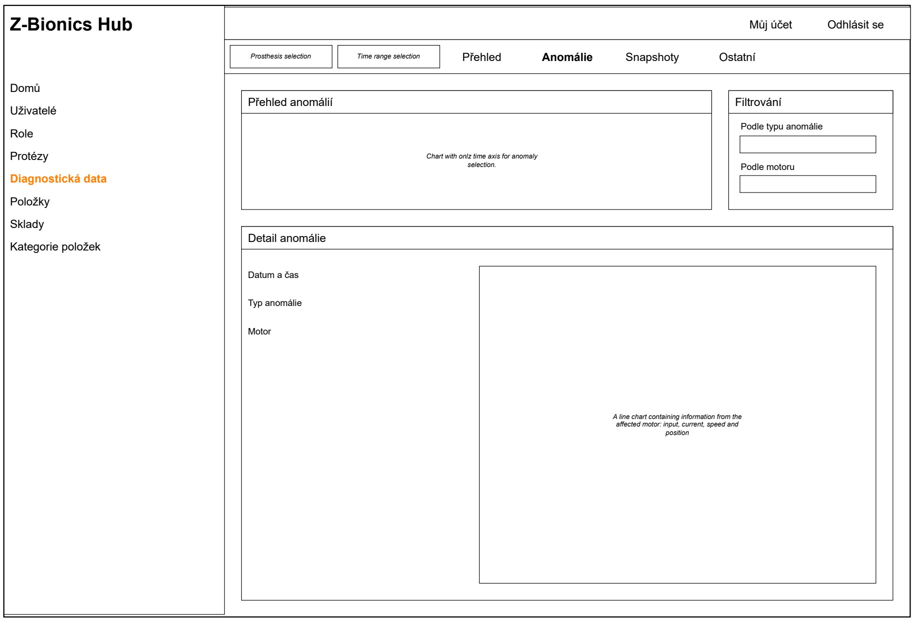

Figure B.2: Wireframe of diagnostics anomalies page

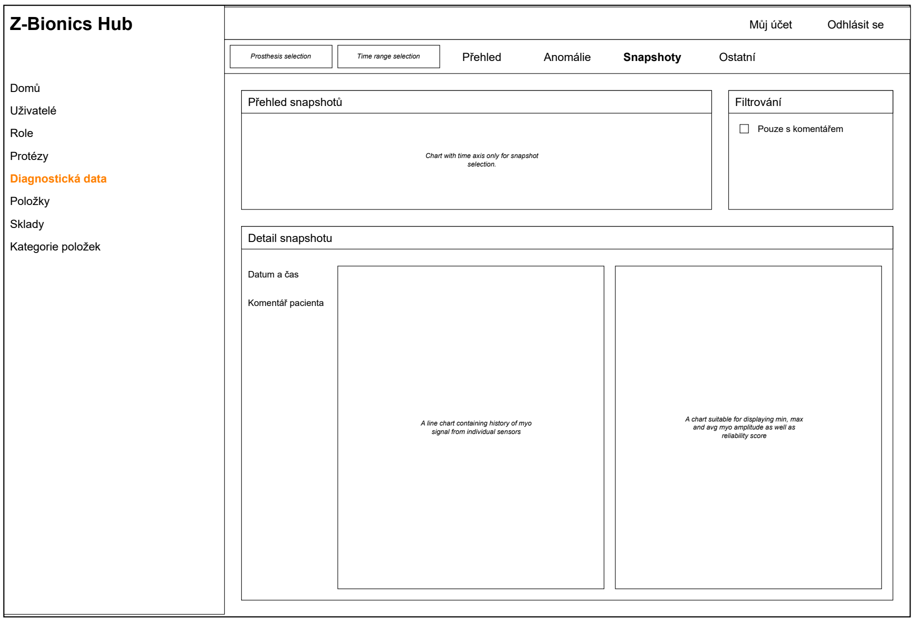

Figure B.3: Wireframe of diagnostics snapshots page

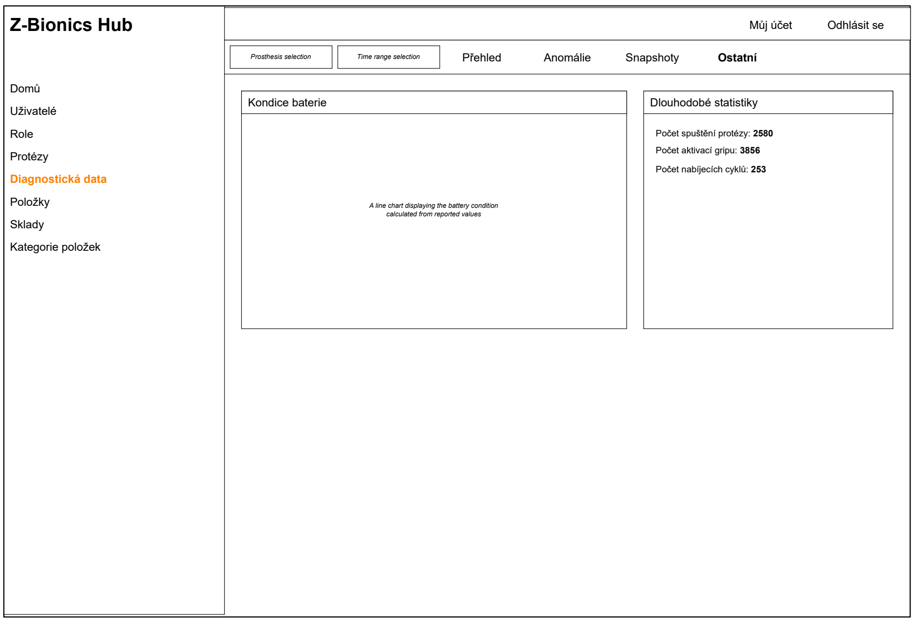

Figure B.4: Wireframe of other information page

Appendix C

Mobile Application Implementation

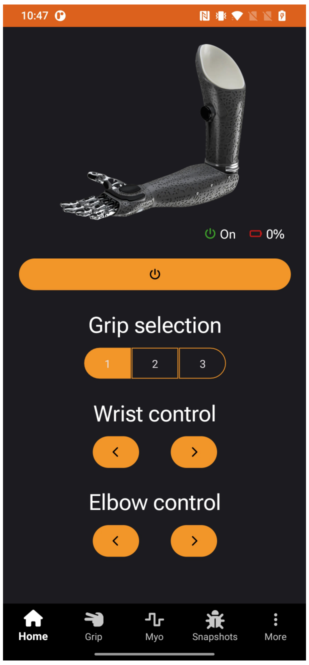

Figure C.1: Mobile application—Main page

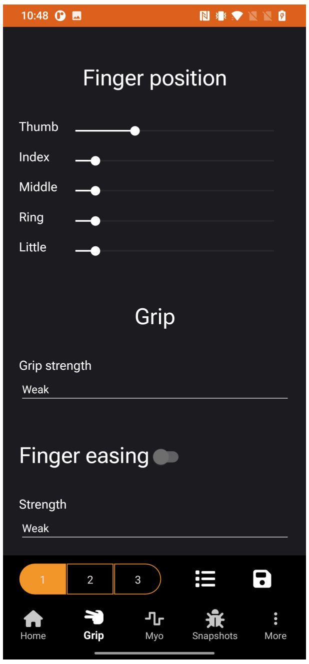

Figure C.2: Mobile application—Grip page

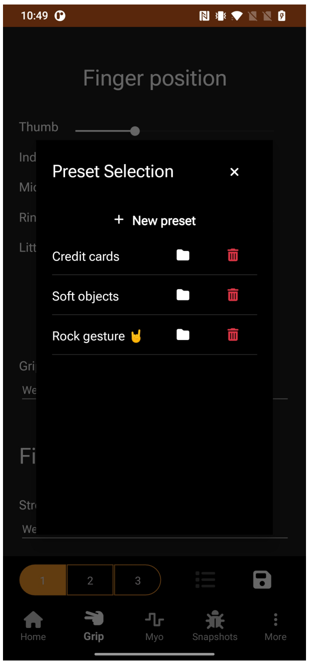

Figure C.3: Mobile application—Grip page with preset selection dialog opened

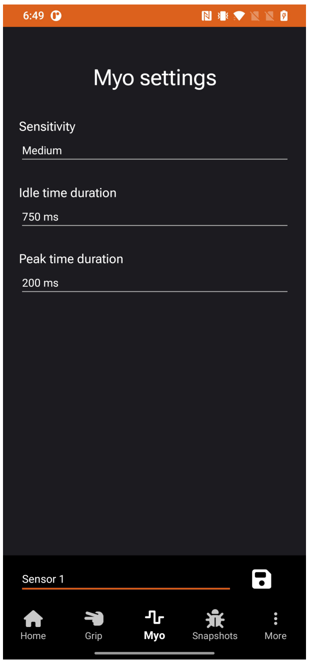

Figure C.4: Mobile application—Myo settings page

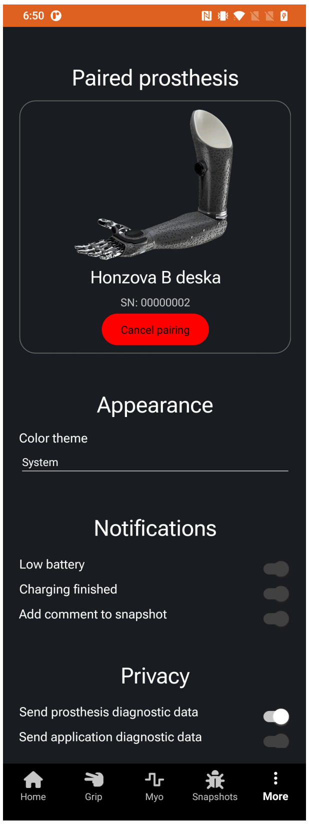

Figure C.5: Mobile application—Application settings page

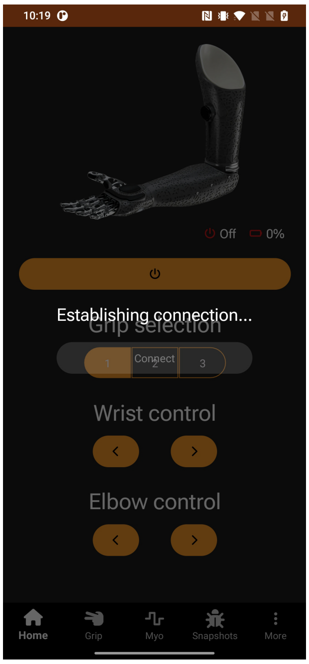

Figure C.6: Mobile application—Overlay when the prosthesis is not connected

Appendix D

Z-Bionics Hub Extension Implementation

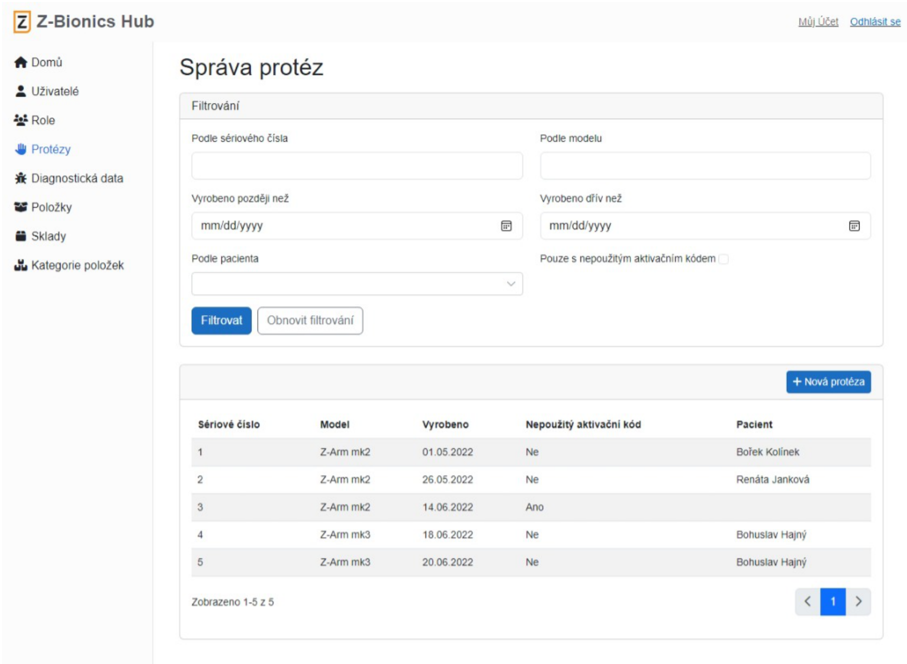

Figure D.1: Z-Bionics Hub—Prosthesis list page

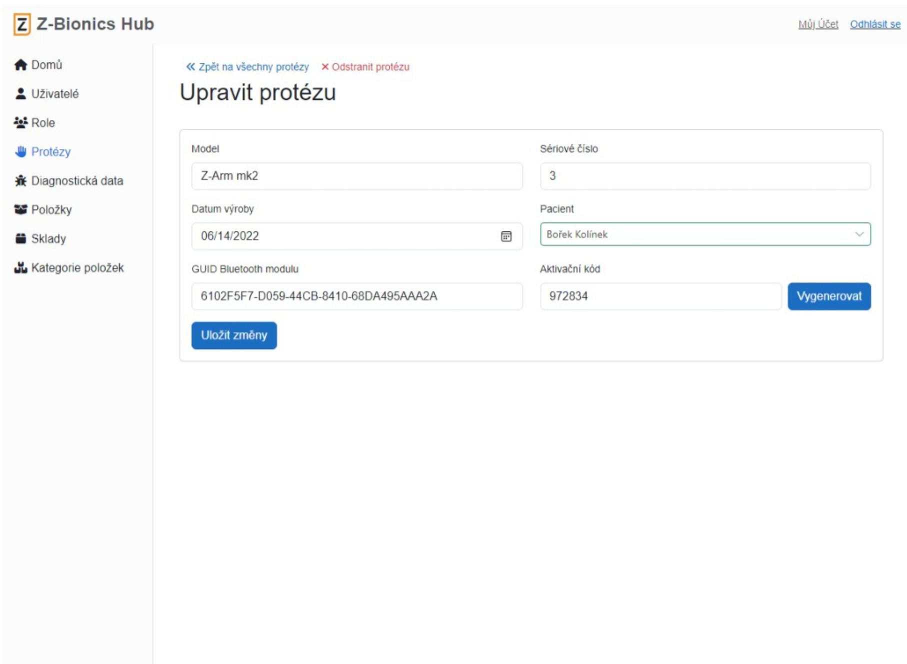

Figure D.2: Z-Bionics Hub—Prosthesis edit page

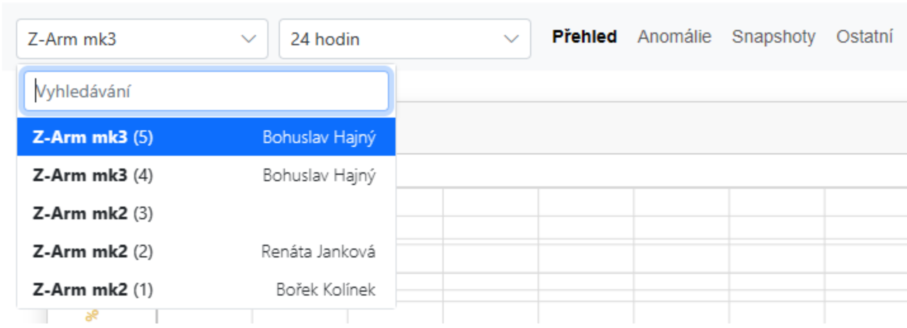

Figure D.3: Z-Bionics Hub—Diagnostics navbar

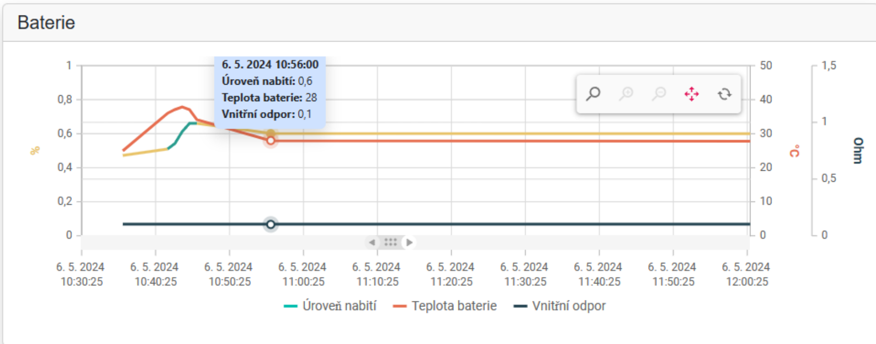

Figure D.4: Z-Bionics Hub—Battery chart

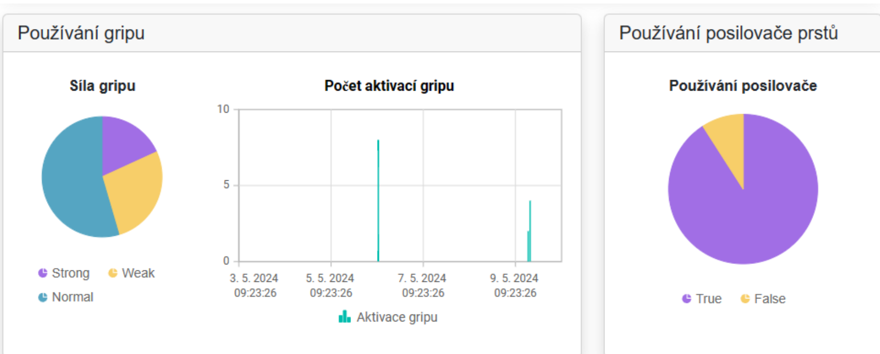

Figure D.5: Z-Bionics Hub—Grip usage charts

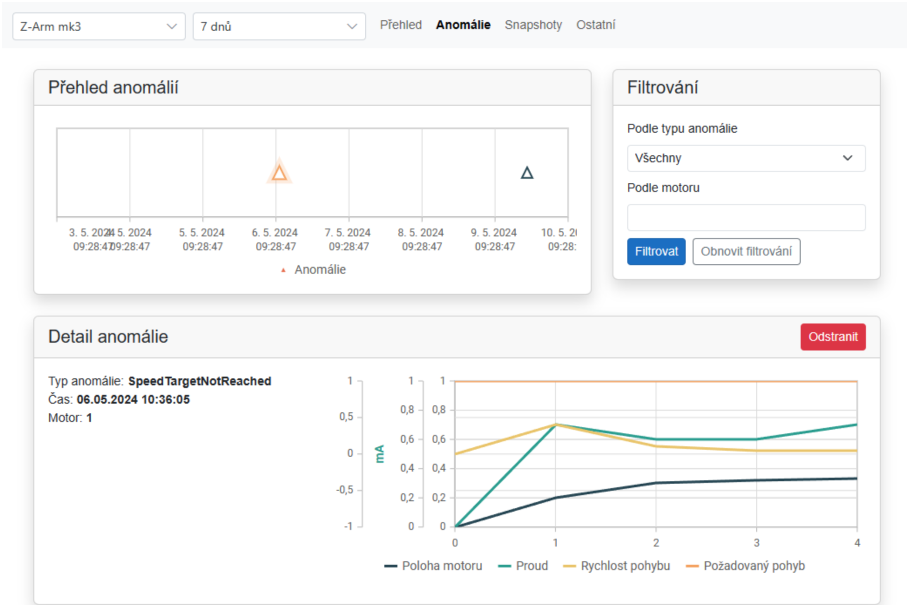

Figure D.6: Z-Bionics Hub—Anomaly component

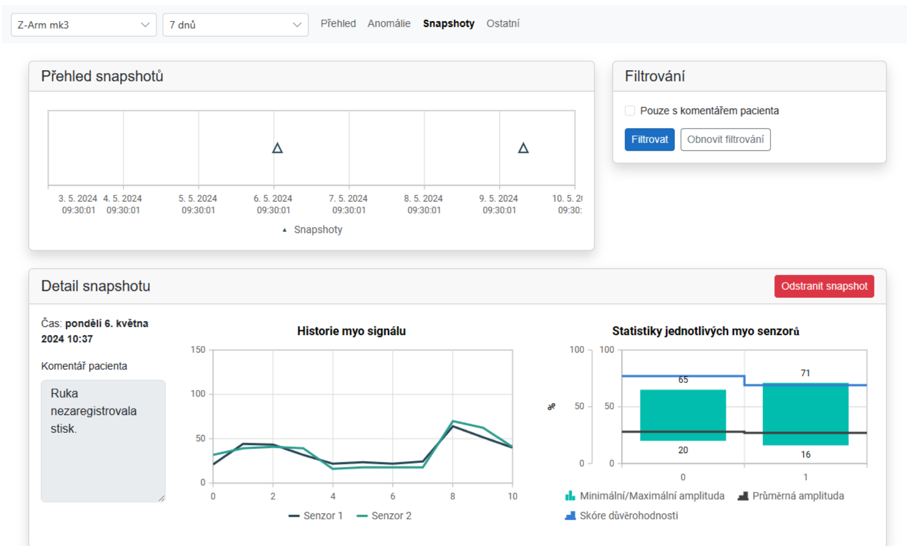

Figure D.7: Z-Bionics Hub—Snapshot component