

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Aplikace pro zpracování a zobrazení Elektromyografického
signálu



2023

Vedoucí práce:
doc. RNDr. Michal Krupka Ph.D.

Jan Vychodil

Studijní program: Informatika,
Specializace: Programování a vývoj
software

Bibliografické údaje

Autor: Jan Vychodil
Název práce: Aplikace pro zpracování a zobrazení Elektromyografického signálu
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2023
Studijní program: Informatika, Specializace: Programování a vývoj software
Vedoucí práce: doc. RNDr. Michal Krupka Ph.D.
Počet stran: 30
Přílohy: elektronická data v úložišti katedry informatiky
Jazyk práce: český

Bibliographic info

Author: Jan Vychodil
Title: Application for processing and representation of Electromyographic signal
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2023
Study program: Computer Science, Specialization: Programming and Software Development
Supervisor: doc. RNDr. Michal Krupka Ph.D.
Page count: 30
Supplements: electronic data in the storage of department of computer science
Thesis language: Czech

Anotace

Cílem této bakalářské práce je vytvoření desktopové aplikace pro měření elektromyografického signálu (EMG). EMG signál je převeden na vstup zvukové karty počítače a následně matematicky zpracován. Aplikace obsahuje konfiguraci jednotlivých měření pro různé pacienty, a následné sledování průběhu měření v reálném čase. Naměřená data a údaje pacientů jsou ukládány do databáze. Je možné zpětné procházení naměřených dat, a jsou k dispozici nástroje pro úpravu databáze pacientů.

Synopsis

The aim of this bachelor's thesis is to create a desktop application for measuring electromyographic signals (EMG). The EMG signal is amplified, passed to the input of the computer's sound card and then processed mathematically. The application includes configuration of measurements setup for different patients, and subsequent real-time display of the measurement progress. The measured data and patient information are stored in the database. It is possible to retrospectively review the measured data, and there are available some tools for editing the patient database.

Klíčová slova: desktopová aplikace; zpracování elektromyografického signálu; Windows App SDK; WinUI3; SQLite

Keywords: desktop app; electromyography signal processing; Windows App SDK; WinUI3; SQLite

Děkuji vedoucímu práce doc. RNDr. Michalu Krupkovi Ph.D za užitečné připomínky a rady na konzultacích.

Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

Obsah

1	Úvod	8
1.1	Cíle práce	8
2	Zpracování EMG signálu	9
2.1	EMG signál	9
2.2	Cesta analogového EMG signálu	9
2.3	Digitalizace signálu	9
2.4	Posuvné okno	10
2.5	Výpočet frekvenčního spektra	10
2.6	Filtrace frekvencí	11
2.7	Získání dominantních frekvencí	11
2.7.1	Výpočet střední hodnoty diskrétní náhodné veličiny	11
2.7.2	Výpočet mediánu diskrétní náhodné veličiny	12
2.8	Klouzavý průměr	13
2.9	výpočet regresní přímky	13
3	Uživatelská příručka	14
3.1	Navigace	14
3.2	Konfigurace	14
3.3	Měření	15
3.4	Pacienti	16
3.5	Záznamy	18
3.6	Nastavení	18
4	Programátorská dokumentace	19
4.1	Použité technologie a jazyky	19
4.1.1	C#	19
4.1.2	XAML	19
4.1.3	Windows App SDK	19
4.1.4	WinUI 3	20
4.1.5	MSIX Packaging Tool	20
4.2	Použité knihovny	20
4.2.1	SQLite	20
4.2.2	Windows Community Toolkit MVVM	20
4.2.3	NAudio	21
4.2.4	FftSharp	21
4.2.5	Live Charts 2	21
4.3	Struktura Programu	21
4.3.1	Architektura MVVM	21
4.3.2	Ukládání dat	23
	Závěr	24

Conclusions	25
A Obsah elektronických dat	26
Seznam zkratek	27
Literatura	28

Seznam obrázků

1	Výpočet frekvenčního spektra	10
2	Stránka konfigurace	15
3	Stránka měření	16
4	Vyskakovací panel pro přidání svalu	16
5	Stránka pacientů	17
6	Stránka přidání pacienta	17
7	Stránka záznamů	18
8	Diagram databáze	23

1 Úvod

S elektromyografickým signálem (EMG) se standardně pracuje v různých, většinou medicínských oborech. Pro měření zpracovávání a zobrazování se používají specializované a často drahé přístroje. Hlavní myšlenka této práce je, že EMG signál je možné zpracovávat pomocí běžné zvukové karty počítače[1]. Kromě pouhého zobrazování a zaznamenávání se otevírají různé možnosti pro zpracování signálu, zejména pak spektrální analýza. V dnešní době i pomalejší počítačová zařízení nemají s těmito výpočty problém.

Změny dominantních frekvencí frekvenčního spektra souvisejí s únavou svalů a jejich celkovým stavem. Tohoto jevu je využíváno ve sportovní medicíně pro měření trénovanosti svalů, ale také pro měření různých svalových poruch. V poslední době se objevily práce[2], které naznačují vztah mezi změnami dominantních frekvencí a sarkopenií. Sarkopenie je svalové onemocnění vyskytující se zejména u lidí ve vyšším věku a způsobující úbytek svalové hmoty, což vede ke snížení pohyblivosti a ztrátě soběstačnosti. Včasnou diagnostikou je možné tuto nemoc zachytit, což může zlepšit kvalitu života pacientů a snížit náklady na zdravotní a sociální systém. Aplikace pro běžné počítače, která vhodně zpracovává EMG signál, se dá prakticky využít pro testování a plošný screening pacientů bez nutnosti využívání specializovaných pracovišť a přístrojů. Cílem této práce je takovou aplikaci vytvořit.

1.1 Cíle práce

Účelem této práce je vytvoření aplikace, která zpracovává a vyhodnocuje elektromyografický signál svalů jednotlivých pacientů. Aplikace podle zadaných metod zpracovává vstupní EMG signál a hledá jeho dominantní frekvence. Naměřená data jsou v relaci s jednotlivými pacienty a je možné si je zpětně prohlížet. Aplikace je primárně určena pro desktopová zařízení s operačním systémem Windows. Z uživatelského hlediska by aplikace měla obsahovat následující prvky:

- Konfigurace jednotlivých měření
- Výběr pacienta a svalů, které chceme měřit
- Při měření v reálném čase se graficky zobrazuje frekvenční spektrum EMG signálu a vývoj jeho dominantní frekvence
- Po skončení měření se graficky zobrazí regresní přímka z grafu dominantních frekvencí a její číselné hodnoty, t.j. počáteční frekvence a sklon přímky
- Seznam pacientů a jejich jednotlivých měření
- Možnost přidávat, mazat a editovat jednotlivé pacienty
- Prohlížení záznamů starších měření

2 Zpracování EMG signálu

V této sekci je stručně posán celý proces zpracování EMG signálu, od snímání signálu ze svalu po data připravená k zobrazení. Metody pro zpracování jsou čerpány z literatury[3][4] a tato práce se zabývá jejich programovou implementací.

2.1 EMG signál

Elektromyografický signál vzniká v důsledku elektrické aktivity svalových buněk. Svalové vlákno při své kontrakci vytváří slabé elektrické napětí, tzv. akční potenciál. Když snímáme EMG signál povrchovými elektrodami, výsledný signál je časovým a prostorovým součtem mnoha akčních potenciálů z jednotlivých svalových vláken. Tyto signály se komplikovaným způsobem sčítají a mají charakteristiku náhodné veličiny.

2.2 Cesta analogového EMG signálu

EMG signál je snímán pomocí tří povrchových elektrod, dvě aktivní, třetí slouží jako uzemnění. Elektrody jsou umístěny na měřený sval, jejich přesná pozice je určena odborníkem. Maximální hodnoty amplitud snímaného signálu se pohybují okolo 1 mV. Pro přesné zpracování signálu zvukovou kartou PC jsou tyto hodnoty příliš nízké. Proto, a z důvodu vlastností bioelektrického signálu, je signál veden z elektrod do diferenciálního zesilovače, který jej zesílí zhruba tisíckrát. Zesilovač je napájen pomocí USB a jeho výstupní signál je napojen do zvukové karty PC standardním konektorem typu *Jack*. Hlavním účelem aplikace je hledání dominantních frekvencí EMG signálu. Ty se pohybují v rozmezí 40 Hz a 250 Hz. Standardní zvuková karta dokáže zpracovávat signál, minimálně v rozsahu 20 Hz až 20 KHz. Její využití pro zpracování EMG signálu je tedy technicky možné.

2.3 Digitalizace signálu

Vstupní spojitý EMG signál musíme převést do digitální podoby. Digitalizace se obecně skládá ze dvou kroků. Nejprve je nutné spojitý signál rozdělit na jednotlivé vzorky. Tato fáze, nazývaná vzorkování, probíhá v časových intervalech určených vzorkovacími frekvencemi. Vzorkovací frekvence se udává v jednotkách Hz a určuje, kolik bude ze signálu vzato vzorků za sekundu. Druhá fáze je tzv. kvantování, které jednotlivým vzorkům přiděluje jejich hodnotu (sílu signálu). Digitalizaci provádí A/D převodník, který se v počítači nachází ve zvukové kartě.

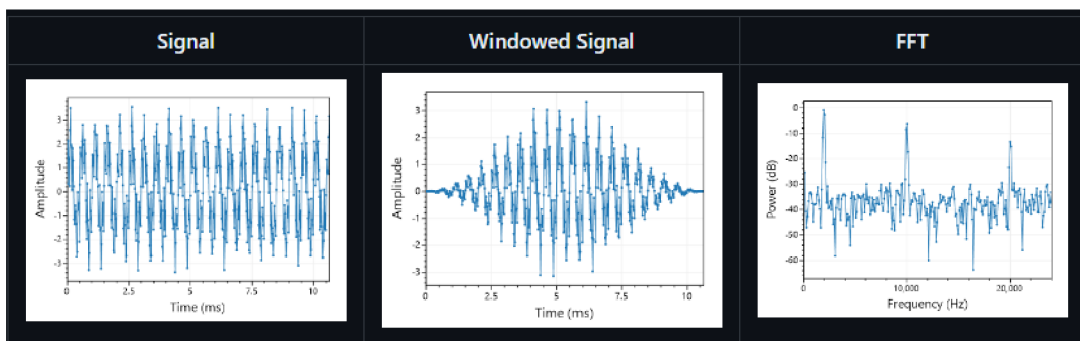
Pro tuto práci bude hodnota rozlišení A/D převodníku nastavena na 16bit, počet kvantizačních hladin bude 2^{16} . Vzorkovací frekvence musí splňovat Shannonův teorém, tj. že vzorkovací frekvence se rovná minimálně dvojnásobku nejvyšší frekvence, která se v daném vzorkovaném signálu vyskytuje. Vzorkovací frekvence bude v této práci nastavitelná hodnota. Vezmeme-li v potaz Shannonův teorém a fakt, že použitý zesilovač je fyzicky limitován na 500 Hz, ustanovíme minimální nastavitelnou hodnotu vzorkovací frekvence na 1000 Hz.

2.4 Posuvné okno

Hlavním účelem aplikace je zjišťování dominantních frekvencí měřeného EMG signálu a jejich vývoj v čase. Při aktivním měření bude digitalizovaný EMG signál tvořit proud dat (vzorků). Po těchto vzorcích budeme posouvat klouzavé okno. Při každém posunu okna se ze vzorků, nacházejících se uvnitř okna, vypočítá frekvenční spektrum a aktuální dominantní frekvence. Šířka a rychlost posunu okna jsou v aplikaci nastavitelné proměnné.

2.5 Výpočet frekvenčního spektra

Pro výpočet frekvenčního spektra existuje metoda Diskrétní Fourierovy transformace, jejíž algoritmus má časovou složitost $O(n^2)$). Pro praktické použití byl vyvinut efektivnější algoritmus: FFT (Fast Fourier Transform). Ten má časovou složitost $O(n \cdot \log_2 n)$. Matematicky Fourierova transformace počítá s nekonečným počtem vzorků. Pro účely programu je ale nutné aplikovat algoritmus FFT na okno o konečném počtu vzorků. To by způsobovalo vytváření parazitních frekvencí. K odstranění tohoto problému se využívají funkce, které na konci a začátku okna vytvoří postupný přechod hodnot k nule. Jedna z nejčastějších metod je tzv. Hammingovo okno, které zde bude využito.



Obrázek 1: Výpočet frekvenčního spektra

2.6 Filtrace frekvencí

Protože měřený signál může obsahovat mnohé rušení, je před výpočtem dominantních frekvencí provedena filtrace. Konkrétní frekvence je možné jednoduše filtrovat přímo na frekvenčním spektru. Každý prvek frekvenčního spektra popisuje hodnotu síly určité frekvence - pro potlačení je možné ji nastavit na "0". Největší rušivý signál bývá z elektrické sítě, který se pohybuje okolo 50 Hz nebo 60 Hz. Signál filtrujeme třemi způsoby:

- $\text{LowPassFilter}(f)$ — potlačí všechny frekvence vyšší než f
- $\text{HighPassFilter}(f)$ — potlačí všechny frekvence nižší než f
- $\text{NotchFilter}(f)$ — potlačí konkrétní frekvenci f

2.7 Získání dominantních frekvencí

Spektrum EMG signálu má charakteristiku náhodné veličiny, je *rozeklané*. Pro zjištění dominantní frekvence nestačí pouze zjistit nejvyšší hodnotu, je nutné použít statistické metody. V aplikaci je možnost výběru jednoho ze dvou způsobů výpočtu.

2.7.1 Výpočet střední hodnoty diskrétní náhodné veličiny

Střední hodnotu M , počítáme s frekvenčního spektra, které je reprezentováno polem $data$ o velikosti n . Vzorec pro výpočet je:

$$M = \sum_{i=0}^{n-1} f_i \cdot p_i$$

Hodnota f_i reprezentuje hodnotu frekvence i — *tého* prvku v poli $data$

$$f_i = i \cdot \text{spektrální rozlišení} = i \cdot \frac{\text{vzorkovací frekvence}}{\text{velikost okna}}$$

Hodnota p_i reprezentuje pravděpodobnost i — *tého* prvku v poli $data$

$$p_i = \frac{data[i]}{\sum_{j=0}^{n-1} data[j]}$$

Po dosazení a úpravách dostaneme následující vzorec

$$M = \frac{\text{vzorkovací frekvence}}{\text{velikost okna}} \cdot \frac{1}{\sum_{j=0}^{n-1} data[j]} \cdot \sum_{i=0}^{n-1} i \cdot data[i]$$

Výpočet tohoto vzorce je v programu implementován odpovídajícím algoritmem.

2.7.2 Výpočet mediánu diskrétní náhodné veličiny

Tato druhá metoda je popsána na konkrétním algoritmu z programu aplikace.

Zdrojový kód 1: algoritmus pro výpočet mediánu diskrétní náhodné veličiny

```
1 private double CalculateMedianValue(MeasurementGroup m, double[]
  data)
2 {
3     double sum = 0;
4     double probabilitySum = 0;
5     double lowIndex = 0;
6     double deltaHigh = 0;
7     double deltaLow = 0;
8     for (var i = 0; i < m.SpectrumDataSize; i++)
9     {
10        sum += data[i];
11    }
12    for (var i = 0; i < m.SpectrumDataSize; i++)
13    {
14        probabilitySum += data[i] / sum;
15        if (probabilitySum > 0.5 && i > 0)
16        {
17            lowIndex = i - 1;
18            deltaHigh = probabilitySum - 0.5;
19            deltaLow = 0.5 - (probabilitySum - (data[i] / sum));
20            break;
21        }
22    }
23    var result = (lowIndex + (deltaLow / (deltaHigh + deltaLow))
24 ) * m.SpectralResolution;
25    return result;
26 }
```

Zdrojový kód 2: Algoritmus pro výpočet diskrétní náhodné veličiny

Parametr metody `m` obsahuje konfiguraci pro aktuální měření. Parametr `data` představuje hodnoty frekvenčního spektra ze kterých počítáme medián. Proměnná `sum` reprezentuje součet všech hodnot frekvenčního spektra. Druhý cyklus prochází celé pole `data` a při každé iteraci přičte do proměnné `probabilitySum` hodnotu pravděpodobnosti výskytu i - tého prvku. Medián se vyskytuje tam, kde proměnná `probabilitySum` se rovná hodnotě 0.5. Jelikož pracujeme s diskrétní veličinou, rovnost většinou nenastane. Když proměnná `probabilitySum` přesáhne hodnotu 0.5, medián se bude vyskytovat mezi hodnotami frekvenčního indexu i a $i - 1$. Proměnná `deltaHigh` je odchylka sumy pravděpodobností na indexu i od sumy pravděpodobností odpovídající skutečnému mediánu (0.5). `deltaLow` je odchylka součtu pravděpodobností na indexu $i - 1$ od sumy pravděpodobností odpovídající skutečnému mediánu. Přesnou hodnotu mediánu vypočítáme s pomocí metody lineární interpolace, řádek 23.

2.8 Klouzávý průměr

Hodnoty dominantních frekvencí kolísají rychle a chaoticky (systematicky ale klesají). Proto pro účely jejich zobrazení v grafu na ně aplikujeme klouzávý průměr. Klouzávý průměr se aplikuje na hodnoty dominantních frekvencí získané v časovém rámci dvou sekund. Tedy graf dominantních frekvencí se vždy začne vykreslovat až po uplynutí dvou sekund od doby získané první hodnoty dominantní frekvence. Časové okno o délce 2 s je vhodné, neboť únava svalů se typicky neprojevuje rychleji než za tuto dobu.

2.9 výpočet regresní přímky

Graf hodnot dominantních frekvencí proložíme regresní přímkou a zaznamenáme dva její důležité parametry. Lineární přímkou vyjádříme rovnicí:

$$y = a \cdot x + b$$

- Hodnota a reprezentuje sklon regresní přímky
- Hodnota b reprezentuje počáteční hodnotu regresní přímky

Tyto hodnoty vypočítáme s pomocí následujících vzorců:

$$a = \frac{\sum_{i=0}^{n-1} [(x[i] - \bar{x}) * (y[i] - \bar{y})]}{\sum_{i=0}^{n-1} (x[i] - \bar{x})^2}$$

$$\bar{x} = \frac{1}{n+1} \cdot \sum_{i=0}^{n-1} x[i]$$

$$\bar{y} = \frac{1}{n+1} \cdot \sum_{i=0}^{m-1} y[i]$$

$$b = \bar{y} - a \cdot \bar{x}$$

3 Uživatelská příručka

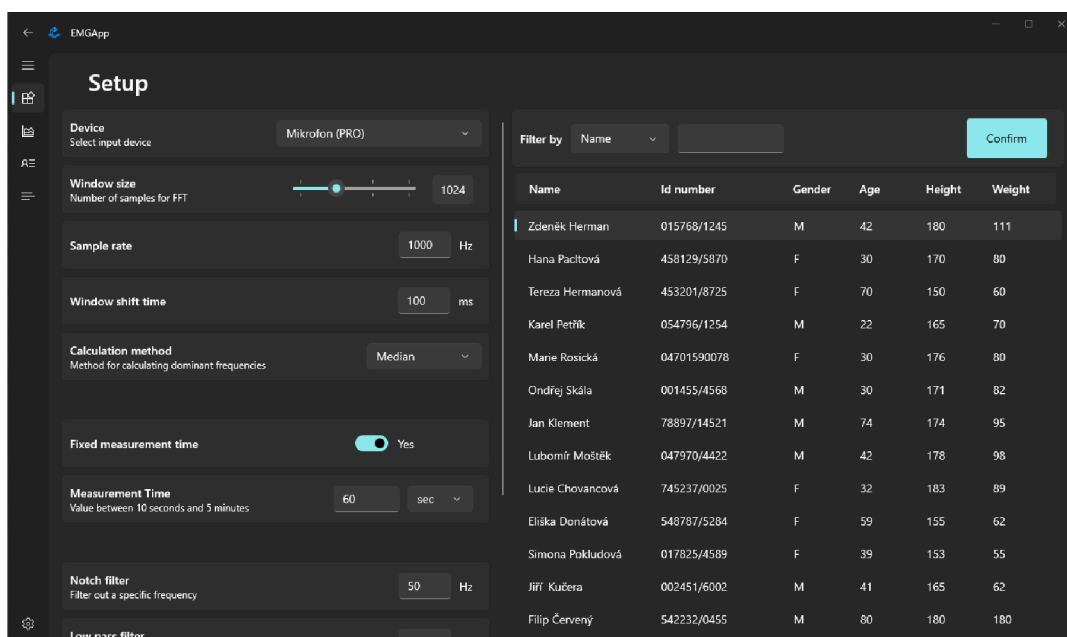
V této části textu je popsána funkcionalita aplikace z pohledu uživatele. Aplikace je navržena tak, aby byla intuitivní, přehledná, a také aby mohla být používána i uživatelem, který nemá konkrétní technické znalosti o způsobech zpracování EMG signálu.

3.1 Navigace

Obsah aplikace je zobrazován na jednotlivých stránkách, mezi nimi se navigujeme převážně pomocí navigačního panelu. Stránky přístupné z navigačního panelu jsou: [konfigurace](#), [měření](#), [pacienti](#), [záznamy](#) a [nastavení](#). Dále jsou zde stránky pro přidání a editaci pacienta, které jsou však přístupné pouze s pomocí uživatelských prvků na stránce [pacienti](#). Podrobněji jsou stránky popsány v následujících kapitolách. Samotný navigační panel má tři módy, které se mění v závislosti na velikosti celé aplikace.

3.2 Konfigurace

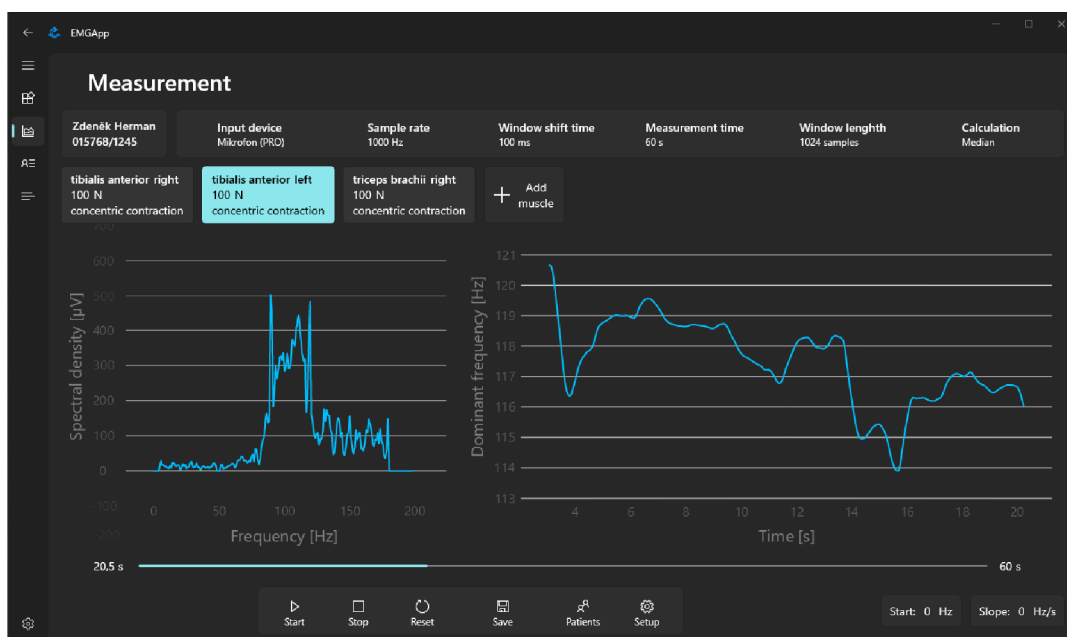
Tato stránka slouží jako první krok před spuštěním měření. Je rozdělena na dvě části. Na pravé straně se nachází podrobné nastavení měření. Vždy je nutné zkontrolovat kartu pro výběr vstupního zařízení. V úvodu kapitoly je uvedeno, že aplikaci může používat i člověk bez technických znalostí zpracování signálu. Výjimkou je toto podrobné nastavování konfigurace, nicméně, neočekává se příliš časté provádění těchto změn z důvodu obvyklého postupu provádět série měření s předem ustálenou konfigurací. V levé části obrazovky je seznam pacientů, ze kterého vybereme pacienta pro měření. Pro usnadnění nalezení pacienta je možné využít jeden z filtrů, který se nachází v panelu nad seznamem pacientů. Pro potvrzení nastavení je nutné zmačknout tlačítko nacházející se vpravo od filtru. Po kliknutí na tlačítko dojde ke změně aktuálně měřeného kontextu a přesměrování na obrazovku pro [měření](#).



Obrázek 2: Stránka konfigurace

3.3 Měření

Na této stránce je nejdůležitější funkcionalita celé aplikace. V horní části je ve dvou panelech zobrazen kontext aktuálního měření, a to základní informace o pacientovi a důležité hodnoty konfigurace. Pod tímto panelem se nachází oblast, ve níž jsou umístěny konkrétní měřené svaly, a tlačítko pro přidání jednotlivého svalu. Kliknutím na toto tlačítko se zobrazí vysouvací panel, ve kterém vybereme měřený sval a vyplníme další informace týkající se jeho měření. Svalů můžeme přidat libovolné množství, ale vždy se musí jednat o různé druhy svalů. Mezi svaly je možné libovolně přepínat. Strukturálně pod kartami svalů jsou umístěny dva grafy. Levý graf zobrazuje spektrální analýzu měřeného signálu. Pravý graf zobrazuje vypočítané hodnoty dominantních frekvencí, pro které je po skončení měření vykreslena v grafu regresní přímka. Pod grafy se nachází ukazatel průběhu měření. Ovládací panel je umístěn ve spodní části stránky a obsahuje tlačítka pro spuštění a resetování měření. Dále tlačítko pro uložení všech dat a tlačítka pro rychlý přechod do stránky [konfigurace](#) a [pacientů](#). Dva důležité parametry regresní přímky dominantních frekvencí jsou umístěny právě dolní části stránky.



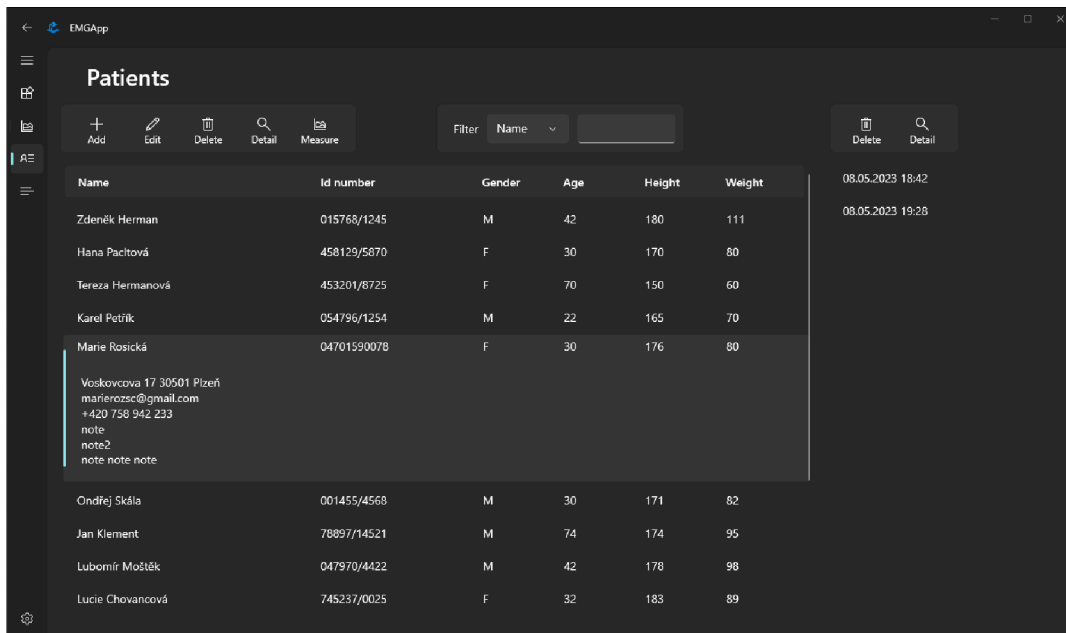
Obrázek 3: Stránka měření

Obrázek 4: Vyskakovací panel pro přidání svalu

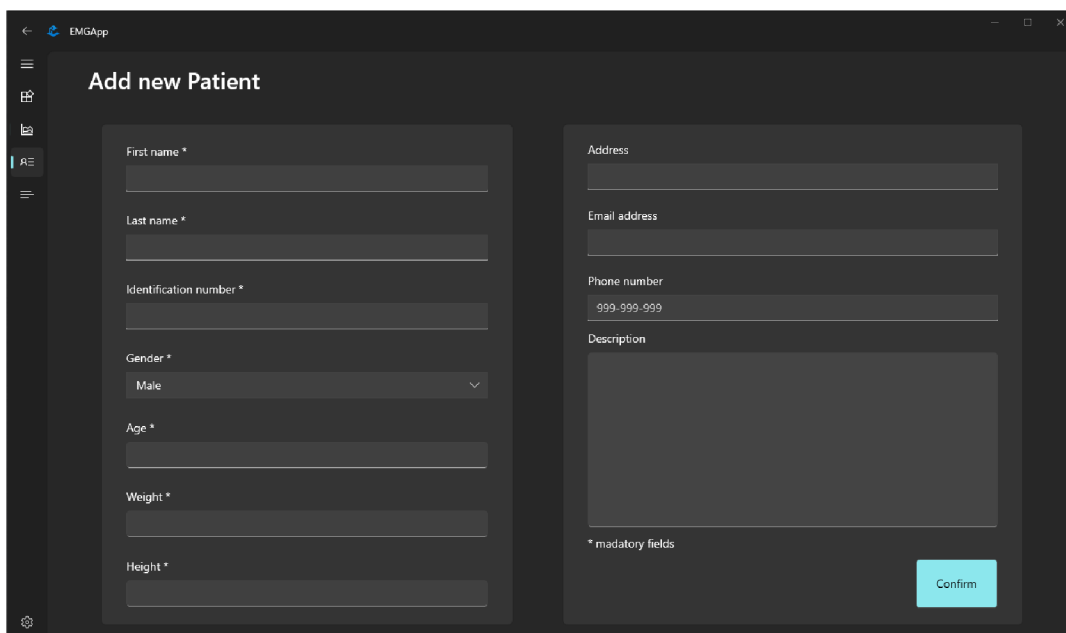
3.4 Pacienti

Většinu další stránky vyplňuje tabulka pacientů a jejich jednotlivých měření. Nad tabulkou pacientů je umístěn ovládací panel, který obsahuje tlačítka pro přidávání, editování a mazání pacientů. Dále tlačítko pro zobrazení detailních informací pacienta, a tlačítko pro vytvoření nového měření s defaultními konfiguračními hodnotami a vybraným pacientem. Po kliknutí na tlačítka přidání a editace pacienta je uživatel přesměrován do jiné stránky, která slouží k tomuto účelu. Také je zde panel pro filtrování pacientů, který je stejný jako na stránce

konfigurace. Pro tabulku měření je zde také ovládací panel, který umožňuje vybraný záznam smazat nebo zobrazit jeho detail, tato akce přesměruje uživatele do stránky [záznamů](#).



Obrázek 5: Stránka pacientů



Obrázek 6: Stránka přidání pacienta

3.5 Záznamy

Stejně jako na stránce měření jsou v horní části umístěny panely zobrazující aktuální kontext, a navíc datum pořízení záznamu. Pod nimi se nachází tabulka se svaly, které byly měřeny v rámci tohoto záznamu, a poté tři grafy. Obsah prvního grafu je vykreslený hned při navigaci na stránku a je tvořen hodnotami dominantních frekvencí a regresní přímkou. Další graf zobrazuje čistě snímaný signál a poslední graf zobrazuje hodnoty spektrální analýzy. Měření lze přehrávat s pomocí tlačítek na ovládacím panelu který je umístěn v dolní části stránky. Nad ovládacím panelem se nachází ukazatel průběhu přehrávaného měření, který je interaktivní a lze jím manuálně pohybovat.



Obrázek 7: Stránka záznamů

3.6 Nastavení

Tato stránka obsahuje informace a nastavení, které se týká aplikace jako celku. Jsou zde tři možnosti pro výběr motivu aplikace, ten může být světlý, tmavý nebo defaultní. Defaultní znamená že aplikace použije mód, který aktuálně využívá UI operačního systému. Pokud je aplikace spuštěna na operačním systému Windows 11, dokáže také využívat mód průsvitnosti, ačkoliv některé prvky uživatelského rozhraní ho nepodporují. Také se zde vyskytuje pole pro výběr jazyka, které je ale aktuálně limitován pouze na jednu možnost a to angličtinu.

4 Programátorská dokumentace

V této části práce jsou popsány použité technologie a knihovny, proč byly zvoleny a jaký mají účel. Také se zde zaměřuji na popis struktury aplikace a řešení ukládání dat.

4.1 Použité technologie a jazyky

Při výběru technologií jsem vycházel ze zadaných požadavků a vlastních zkušeností. Aplikace by měla být primárně zaměřena na počítače s operačním systémem Windows. V minulosti jsem napsal několik menších aplikací s pomocí jazyka C# a python. Python je velmi všestranný jazyk ale desktopové aplikace nejsou jeho primární zaměření. Navíc v aplikaci musíme v reálném čase zpracovávat velkém množství dat a jazyk Pythone je obecně pomalejší než C#, a také je v něm mnohem jednodušší psát neefektivní kód. Proto jsem zvolil jazyk C# s .NET frameworkem. Pro tento framework mi jsou známé platformy pro vývoj desktopových aplikací Windows Forms a WPF. Windows Forms považuji obecně za zastaralé a nevhodné pro vývoj moderních aplikací. WPF by bylo určitě vhodné řešení, protože je stabilní, osvědčené a existuje pro něj mnoho návodů. Pro vývoj uživatelského rozhraní bych určitě potřeboval další externí knihovnu. Navíc jsem chtěl vyzkoušet platformu která je prezentována jako nejmodernější a bude pravděpodobně ještě dlouhou dobu podporována, proto jsem zvolil platformu WinUI3 (Windows App SDK). Všechny použité technologie a knihovny spadají pod licenci MIT.

4.1.1 C#

C# Je objektově orientovaný jazyk vyvinutý společností Microsoft. Je využíván pro vývoj desktopových, mobilních i webových aplikací. Pro spuštění aplikací se využívá platforma .NET. Aplikace této bakalářské práce využívá .NET verze 6.0.

4.1.2 XAML

XAML (Extensible Application Markup Language) je značkovací jazyk sloužící k popisu uživatelského rozhraní. Jeho hlavním účelem je zjednodušení vývoje uživatelského rozhraní, vše co popisuje jazyk xaml lze napsat i v jazyku c#.

4.1.3 Windows App SDK

Windows App SDK[5] (Windows app software development kit) je vyvinuta společností Microsoft. Jedná se aktuálně o nejnovější sadu knihoven a nástrojů k vývoji desktopových aplikací pro platformy Windows. Jedna z hlavních vlastností je zajištění vizuální a funkcionální konzistence napříč různými operačními systémy windows, od verze Windows 10 (1809), bez nutnosti změn v kódu.

4.1.4 WinUI 3

WinUI 3[6] (Windows User Interface Library) je součástí Windows App SDK. Jedná se o framework poskytující nástroje pro vytváření nativního uživatelské rozhraní pro desktopové Windows aplikace. Prvky uživatelského prostředí využívají Fluent 2[7] design a zajišťují tak vizuální konzistenci s operačním systémem. WinUI3 se nachází stále v aktivním vývoji, proto některé prvky můžou nemusí být zcela implementovány. Pro vývoj je praktické využít aplikaci WinUI 3 Gallery[8], která popisuje všechny dostupné nástroje a ukazuje jejich použití na příkladech.

4.1.5 MSIX Packaging Tool

MSIX[9] (Microsoft Installer XML) je nástroj pro distribuci a instalaci aplikací pro operační systémy Windows 10 a vyšší. MSIX slouží k zapouzdření aplikace a jejich závislostí do izolovaného prostředí. Zlepšuje tak bezpečnost, zaručuje čisté odinstalování aplikace a efektivněji využívá místo na disku.

Při použití tohoto nástroje můžeme aplikaci publikovat v obchodě Microsoft Store, nebo používat jednoduchý instalační nástroj (sideloading). Instalovaná aplikace musí být podepsána ověřeným certifikátem, nebo se certifikát musí nainstalovat manuálně. Data aplikace jsou skrytá v izolovaném prostředí, pokud se k těmto datům ptřebujeme dostat nalezneme je v příslušném adresáři který se nachází v `/Users/<User>/AppData/Local/Packages`.

4.2 Použité knihovny

V této podkapitole jsou popsány Knihovny které jsem potřeboval pro řešení konkrétních problémů během vývoje aplikace.

4.2.1 SQLite

SQLite[10] (Structured query language) je open source relační databázový systém, napsaný v jazyce C. Základní vlastnosti jsou jednoduchost rychlost a spolehlivost. Databáze neobsahuje konfiguraci. SQLite nevyužívá architekturu server—klient, je uložena pouze v jednom souboru k tomuto souboru je přístupováno přes jednoduché api přímo v programu. Čtení, zápis a ukládání dat s použitím SQLite je v průměru efektivnější[11] než standartní metody pro práci s daty na disku.

4.2.2 Windows Community Toolkit MVVM

Toolkit MVVM[12] je Knihovna akceleruje a zjednodušující vývoj aplikací s architekturou MVVM. Je vytvořena a spravována společností Microsoft. Používá se v mnoha oficiálních aplikacích jako například Microsoft Store.

Použití této knihovny významně snižuje objem ručně psaného kódu. Metoda nebo

proměnná třídy(field) je označena příslušným syntaxem, jejich třída je označena klíčovým slovem `partial` a knihovna pak napozadí vygeneruje vygeneruje potřebný kód.

4.2.3 NAudio

NAudio[13] je Open source knihovna pro .NET. Poskytuje mnoho nástrojů pro zpracování audia. V této práci ji využívám pouze pro digitalizaci vstupního signálu na vstupu zvukové karty.

4.2.4 FftSharp

FftSharp[14] je open source knihovna pro .NET. V programu aplikace ji využívám hlavně metody pro efektivní zpracování dat, pomocí FFT (Fast Fourier transform) algoritmu. V programu jsou použity její metody pro výpočty FFT a Hammingova okna.

4.2.5 Live Charts 2

Live Charts 2[15] je open source knihovna pro vytváření grafů a dalších grafických prvků uživatelského rozhraní.

4.3 Struktura Programu

Program byl vytvořen jako projekt v prostředí Visual Studio. Při vytváření projektu byl použit doplněk Windows Template Studio. Ten slouží k akceleraci vývoje a vytvoří základní strukturu aplikace. Aplikace je tvořena jednotlivými stránkami. Pro navigaci mezi nimi byl zvolen navigační panel, tzv. *Hamburger menu*. Aplikace využívá architekturu MVVM. Pro tento účel je využívána knihovna MVVM Toolkit, která významně snižuje objem ručně psaného kódu. Ukládání dat je řešeno využitím relační databáze SQLite.

4.3.1 Architektura MVVM

Model-View-ViewModel je architektura zaměřující se na oddělení kódu a definicí uživatelského rozhraní od kódu logiky aplikace. Skládá se z tří následujících komponentů.

- Model — popisuje data a služby se kterými aplikace pracuje. Je nezávislý, neobsahuje přímé reference na view/view-model.
- View — s pomocí značkovacího jazyka XAML popisuje vzhled uživatelského rozhraní. Data a příkazy jsou propojeny pomocí datových vazeb s view modelem.
- ViewModel — propojuje view a model. Obsahuje logiku jednotlivých příkazů a udržuje stav hodnot uživatelského rozhraní.

Jak bylo řečeno, aplikace se skládá z jednotlivých stránek, mezi kterými se navigujeme pomocí navigačního panelu, nebo interakcí s některými prvky přímo na stránce. Každá stránka má svůj vlastní view a viewModel které jsou v projektu uloženy ve stejnojmenných adresářích `/Views` a `/ViewModels`. V adresáři `/Models` jsou uloženy jednotlivé modely, které jsou následující:

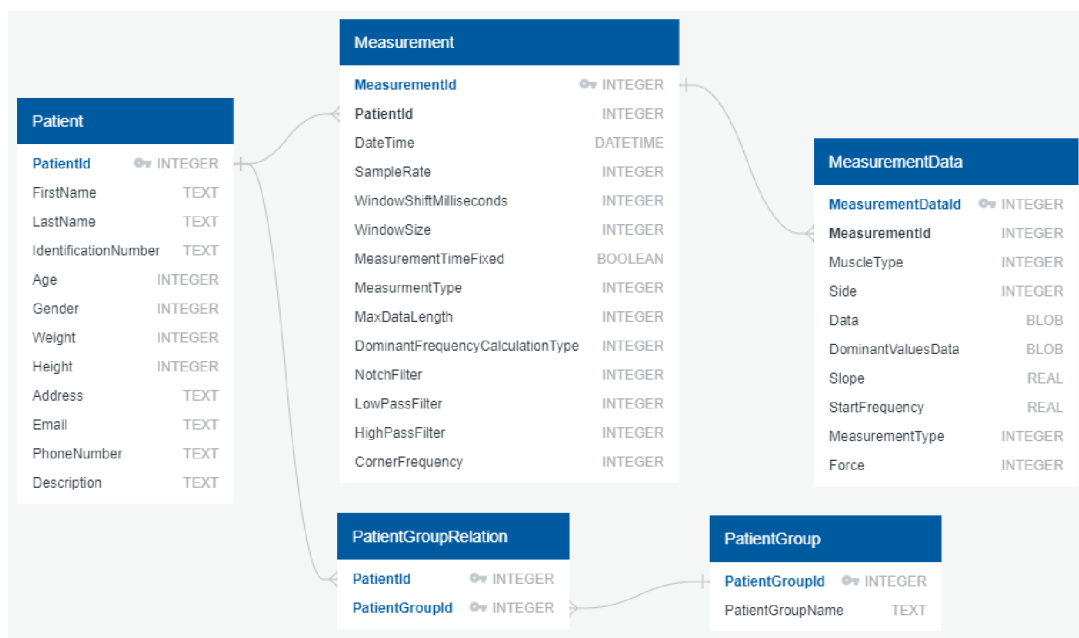
- `Patient` — slouží k uchovávání dat jednotlivých pacientů
- `MeasurementData` — reprezentuje měřený sval, obsahuje jeho typ a vlastnosti a veškerá naměřená data
- `MeasurementGroup` — reprezentuje jednotlivá měření, obsahuje seznam měřených svalů (seznam tříd `MeasurementData`) a hodnoty jejich společné konfigurace

Kromě dlouhodobých dat, která jsou ukládána nebo načítána z databáze, předchozí modely obsahují také pomocné vlastnosti a statická pole která slouží k zjednodušení výpočtů a vizuální reprezentace. Životnost každé stránky je omezena pouze na dobu jejího zobrazení. `ViewModel` tedy nesmí přímo uchovávat žádná trvalá data. Pro Správu modelů a funkcionalitu celé aplikace využívám služby umístěné v adresáři `/Services`. Služby jsou třídy, které jsou zaregistrované při spuštění aplikace, a většinou jsou inicializované jako singleton. Singleton je třída pro kterou je vytvořena maximálně jen jedna instance. `ViewModel` získá jednotlivé služby pomocí dependency injection. To znamená, že reference na služby jsou přidávány a získávány přímo v konstruktoru `ViewModel`. Při vytváření aplikace nástroj `Windows Template Studio` vytvořil služby pro aktivaci aplikace, navigaci mezi stránkami a výběr motivu. Dále jsem pro funkcionalitu programu vytvořil tyto služby:

- `/ MeasurementService.cs` — obsahuje všechny metody zabývající se zpracováním EMG signálu a výpočty dat.
- `/ DatabaseService.cs` — komunikuje s databází, obsahuje obecné metody pro vykonávání příkazů a konkrétní metody které využívají jazyk SQL.
- `/ DataService.cs` — udržuje stavy všech potřebných dat, komunikuje s `DatabaseService.cs` a poskytuje tak rozhraní pro ukládání a načítání dat.

4.3.2 Ukládání dat

Jak už bylo zmíněno pro ukládání dat využívám knihovnu SQLite. Všechna data související s pacienty a jejich měřeními jsou ukládána v jednom databázovém souboru, SQLiteDatabase.db. Tento soubor je uložen v adresáři/LocalState, který je umístěn v izolovaném prostředí aplikace. Program komunikuje s databází SQL příkazy, s pomocí služby DatabaseService.cs. Jednotlivé tabulky, jejich položky a vztahy mezi nimi popisuje následující [obrázek](#). Kromě sloupců popisujících pacienty měření a svaly je potřeba zmínit jaká naměřená data se ukládají. Protože jedna z funkcionalit aplikace je přehrávání starších záznamů z měření, musíme ukládat přímo vzorkovaný vstupní signál. Jeden vzorek má velikost 2 B a pokud použijeme defaultní nastavení, vzorkovací frekvence 1000 Hz a délku měření 60 s, bude jeden naměřený sval potřebovat zhruba 0,12 MB. Při vysokém počtu pacientů a jejich různých měření mohou data zabírat v paměti až jednotky gigabytů. V databázi také ukládáme pole vypočítaných hodnot dominantních frekvencí, jejich velikost vzhledem k počtu ukládaným vzorkům nehraje roli. Pole vzorků je v databázi uloženo ve sloupci Data a pole dominantních frekvencí ve sloupci DominantValuesData. Oba tyto sloupce jsou datového typu BLOB (Binary Large Object). Tento datový typ slouží k ukládání velkých binárních souborů a před uložením se musí provést konverze ukládaných dat na pole bytů. Tabulka PatientGroup a / PatientGroupRelation jsou připravené v databázi pro plánované rozšíření funkcionality aplikace.



Obrázek 8: Diagram databáze

Závěr

Výsledek této práce splňuje zadané cíle. Vytvořená aplikace zpracovává vstupní EMG signál měřených svalů pacientů a výsledky vizuálně zobrazuje. Naměřená data je možné ukládat a zpětně procházet. Aplikace také umožňuje spravovat databázi s pacienty a jejich záznamy o měření.

Funkčnost aplikace byla otestována v praxi a potvrzuje, že aplikované metody pro zpracování signálu fungují správně. [Obrázek 7](#) ukazuje skutečná naměřená data staršího pacienta, kde pokles hodnot v čase značí únavu měřeného svalu.

Na aplikaci by se dalo dále pracovat, v programu by bylo vhodné optimalizovat metody pro práci z databází a také způsoby jakým jsou data počítána. Dále se pak aplikace může rozšiřovat o další funkcionalitu, jako například vytváření skupin, do kterých bychom zařazovaly pacienty podle různých atributů. Zajímavé by také bylo upravit aplikaci, aby mohla měřit více svalů zároveň.

Conclusions

The outcome of this work fulfills the specified requirements. The developed application is processing the input EMG signal of patient muscles and visually displaying the results. The measured data can be saved and reviewed retrospectively. This application also enables the management of the patients records and measurement information database.

The functionality of the application has been tested in practice and the results confirm that the programmed mathematical methods for signal processing are working correctly. The graph in [Figure 7](#) displays measured data from an elderly patient, where the decreasing trend of values over time indicates muscle fatigue.

In the application program, I would like to optimize methods for working with the database and also the ways how the data is computed. Furthermore, the application could be expanded with additional functionality, such as creating groups for categorizing patients based on different attributes. It would be also interesting to modify the application for measuring multiple muscles simultaneously.

A Obsah elektronických dat

src/

Adresář obsahuje veškeré zdrojové kódy v archivu ZIP.

doc/

Adresář obsahuje text práce ve formátu PDF a soubor v archivu ZIP, který obsahuje všechny zdrojové kódy a přílohy nutné k vygenerování tohoto textu.

installation/

Adresář obsahuje soubory potřebné k instalaci aplikace v archivu ZIP.

README.txt

Tento textový soubor obsahuje instrukce pro instalaci a spuštění aplikace.

Literatura

- [1] Crisp, Kevin M. *Recording EMG Signals on a Computer Sound Card*. 2018. Dostupný také z: <https://begol.ir/book/wp-content/uploads/2022/02/june-16-210.pdf>.
- [2] R.Habenicht, G.Ebenbichler. *Age-specific differences in the time-frequency representation of surface electromyographic data recorded during a submaximal cyclic back extension exercise: a promising biomarker to detect early signs of sarcopenia*. 2020. Dostupný také z: <https://doi.org/10.1186/s12984-020-0645-2>.
- [3] Jaroslav, Pospíšil. *Teorie zpracování signálů*. 1994.
- [4] *Pravděpodobnost a statistika*. [online]. [cit. 2023-6-20]. Dostupný z: https://moodle.unob.cz/pluginfile.php/71926/mod_resource/content/1/Studijni_text_3.pdf.
- [5] *Window App SDK*. [online]. [cit. 2023-6-20]. Dostupný z: <https://learn.microsoft.com/en-us/windows/apps/windows-app-sdk/>.
- [6] *WinUI 3*. [online]. [cit. 2023-6-20]. Dostupný z: <https://learn.microsoft.com/en-us/windows/apps/winui/winui3/>.
- [7] *Fluent 2*. [online]. [cit. 2023-6-20]. Dostupný z: <https://fluent2.microsoft.design/>.
- [8] *WinUI 3 Gallery*. 3/29/2021. Dostupný také z: <https://apps.microsoft.com/store/detail/winui-3-gallery/9P3JFPWWDZRC>.
- [9] *MSIX packaging*. [online]. [cit. 2023-6-20]. Dostupný z: <https://learn.microsoft.com/en-us/windows/msix/packaging-tool/tool-overview>.
- [10] *SQLite*. [online]. [cit. 2023-6-20]. Dostupný z: <https://www.sqlite.org/index.html>.
- [11] *SQLite faster than file system*. [online]. [cit. 2023-6-20]. Dostupný z: <https://www.sqlite.org/fasterthanfs.html>.
- [12] *Windows Community Toolkit MVVM*. [online]. [cit. 2023-6-20]. Dostupný z: <https://learn.microsoft.com/en-us/dotnet/communitytoolkit/mvvm/>.
- [13] *NAudio*. [online]. [cit. 2023-6-20]. Dostupný z: <https://github.com/naudio/NAudio>.
- [14] *FftSharp*. [online]. [cit. 2023-6-20]. Dostupný z: <https://github.com/swharden/FftSharp#-fftsharp>.
- [15] *Live Charts 2*. [online]. [cit. 2023-6-20]. Dostupný z: <https://lvcharts.com/docs/winui/2.0.0-beta.850/gallery>.

- [16] *Template Studio for WinUI*. 4/29/2022. Dostupný také z: <https://marketplace.visualstudio.com/items?itemName=TemplateStudio.TemplateStudioForWinUICs>.
- [17] *Visual Studio 2022*. [online]. [cit. 2023-6-20]. Dostupný z: <https://visualstudio.microsoft.com/vs/D>.
- [18] *Biceps Brachii Muscle Fatigue Assessment Through EMG Median Frequency Analysis*. Dostupný také z: https://iceeng.journals.ekb.eg/article_30468_bf47d438a1efcd6efb3c5e346d95018b.pdf.

