



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**WEBOVÁ APLIKÁCIA VÝUKY HRY NA GITARU**

WEB APPLICATION FOR LEARNING HOW TO PLAY THE GUITAR

**BAKALÁRSKA PRÁCA**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MICHAL MIKOTA**

**VEDÚCI PRÁCE**

SUPERVISOR

**Ing. IGOR SZÖKE, Ph.D.**

**BRNO 2023**

## Zadání bakalářské práce



148027

Ústav: Ústav počítačové grafiky a multimédií (UPGM)  
Student: **Mikota Michal**  
Program: Informační technologie  
Specializace: Informační technologie  
Název: **Webová aplikace výuky hry na kytaru**  
Kategorie: Zpracování signálů  
Akademický rok: 2022/23

### Zadání:

1. Analyzujte webové i mobilní aplikace pro výuku hry na kytaru.
2. Navrhněte a implementujte webovou aplikaci pro výuku základů hry na akustickou kytaru s podporou detekce zahraného tónu.
3. Zaměřte se na robustnost vzhledem k akustice prostředí. Aplikaci otestujte na vhodném vzorku beta-testerů.
4. Pokračujte v implementaci a změnách GUI pro co nejlepší UX. Získejte zpětnou vazbu od uživatelů.
5. Zhodnoťte výsledky a navrhněte směry dalšího vývoje.
6. Vyrobte A2 plakátek a nebo cca 30 vteřinové video prezentující výsledky vaší práce.

### Literatura:

- Tidwell et al.: Designing Interfaces: Patterns for Effective Interaction Design, O'Reilly, 2020
- Steve Krug: Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, ISBN: 978-0321965516
- Steve Krug: Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability, ISBN: 978-0321657299
- Joel Marsh: UX for Beginners: A Crash Course in 100 Short Lessons, O'Reilly 2016
- Dále dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:  
Bod 1 a část bodů 2, 3 ze zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Szóke Igor, Ing., Ph.D.**  
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.  
Datum zadání: 1.11.2022  
Termín pro odevzdání: 10.5.2023  
Datum schválení: 31.10.2022

## Abstrakt

Nástroje na výučbu hudobných nástrojov sú tu s nami už stovky rokov a podobne, ako všetko ostatné sa postupne prispôbovali súčasnej dobe. Veľký počet týchto aplikácií je však príliš zložitý, mätúci a vo všeobecnosti pre ich používateľov užívateľsky neprívetivý. Zároveň sa nájde iba zopár aplikácií, ktoré majú implementovanú funkciu zisťovania hraného tónu hráčom v reálnom čase, čo je pre začínajúcich a mierne pokročilých gitaristov veľmi žiaduce a potrebné. Táto bakalárska práca ukazuje a vysvetľuje implementáciu jednotlivých nástrojov potrebných na výučbu pomocou techník detekcie dominantnej frekvencie a filtrovania nežiaducich zvukových artefaktov pomocou algoritmu z oblasti konvolučných neurónových sietí. Zároveň je cieľom sa zamerať na robustnosť algoritmu, t.j. dokáže detekovať a určiť hraný tón na gitare aj v hlučnom prostredí. Jednotlivé nástroje vytvorené v rámci tejto aplikácie využívajú tento algoritmus a v kombinácii s jednoduchým používateľským rozhraním poskytujú možnosť vhodnej alternatívy na výučbu gitarovej hry.

## Abstract

Tools for teaching musical instruments have been with us for hundreds of years and, like everything else, have gradually adapted to the present day. However, a large number of these applications are overly complex, confusing, and generally user-unfriendly for their users. At the same time, there are only a handful of apps that have an implemented feature of real-time detection of the played note by the player, which is very desirable and necessary for beginner and slightly advanced guitar players. This bachelor thesis demonstrates and explains the implementation of the various tools needed for teaching using techniques for dominant frequency detection and filtering out unwanted audio artifacts using an algorithm from the field of convolutional neural networks. At the same time, the goal is to focus on the robustness of the algorithm, i.e., it can detect and determine the played tone on the guitar even in noisy environments. The algorithm used by the individual tools developed within this application, combined with a simple user interface, provide the alternative for teaching how to play guitar

## Klíčové slová

Gitara, Webová, Aplikácia, CREPE, tabulatúra, hudba, tón, artefakt, ladička, nota, akord, m5js, detekcia, frekvencia

## Keywords

Guitar, Web, Application, CREPE, tabulature, music, tone, artifact, tuner, note, chord, m5js, detection, frequency

## Citácia

MIKOTA, Michal. *Webová aplikácia výuky hry na gitaru*. Brno, 2023. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedúci práce Ing. Igor Szöke, Ph.D.

# Webová aplikácia výuky hry na gitaru

## Prehlásenie

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szöke. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Michal Mikota

15. mája 2023

## Podakovanie

Moje podakovanie patrí vedúcemu práce pánovi Ing. Igorovi Szöke Ph.D. za prijatie mojej vlastnej témy bakalárskej práce a za dôsledné vedenie bakalárskej práce a konzultácie, ktoré mi pri práci pomohli.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Prieskum požiadavkov cieľovej skupiny a konkurenčných aplikácií</b>	<b>5</b>
2.1	Súhrn metodík výučby hry na gitaru . . . . .	5
2.2	Cieľová skupina . . . . .	5
2.3	Rozbor aktuálne dostupných webových aplikácií výučby gitarovej hry . . .	7
2.4	Nástroje vyvíjanej aplikácie . . . . .	11
<b>3</b>	<b>Konvolučné neurónové siete (CNN) a algoritmus CREPE</b>	<b>12</b>
3.1	Konvolučné neurónové siete (CNN) . . . . .	12
3.2	CREPE . . . . .	14
3.2.1	Architektúra algoritmu CREPE . . . . .	14
3.3	pYIN a SWIPE . . . . .	16
3.4	Experimentálna časť . . . . .	17
3.4.1	Výsledky vzhľadom na detekciu zahraniého tónu . . . . .	18
3.4.2	Výsledky vzhľadom na akustickú robustnosť prostredia . . . . .	18
<b>4</b>	<b>Nástroje aplikácie</b>	<b>19</b>
4.1	Výber použitých implementačných prostriedkov . . . . .	19
4.2	Knížnica ml5.js a jej využitie v aplikácii . . . . .	19
4.3	Tabulatúra v reálnom čase a jej funkcie . . . . .	21
4.4	Gitarová ladička . . . . .	25
<b>5</b>	<b>Užívateľská prívetivosť v aplikácii</b>	<b>28</b>
5.1	Využitie zásad správneho UX vo webovej aplikácii . . . . .	28
5.2	Užívateľská prívetivosť pre desktopovú verziu . . . . .	30
5.3	Užívateľská prívetivosť pre mobilné zariadenia . . . . .	32
<b>6</b>	<b>Testovanie aplikácie</b>	<b>34</b>
6.1	Alfa testovanie . . . . .	35
6.2	Beta testovanie . . . . .	37
<b>7</b>	<b>Vývoj aplikácie do budúcnosti</b>	<b>39</b>
7.1	Návrh implementácie detekcie akordov vrámci real-time tabulatúry . . . . .	39
7.2	Spôsoby pridávania nových piesní do aplikácie . . . . .	40
7.2.1	Integrácia pomocou API z dostupných webov s databázami tabov . . . . .	40
7.2.2	Rozšírené užívateľské rozhranie pre tvorbu vlastnej tabulatúry . . . . .	41
7.3	Rozšírenie aplikácie pre rôzne typy akustických gitár . . . . .	42

<b>8 Závěr</b>	<b>43</b>
<b>Literatúra</b>	<b>44</b>
<b>A Obsah priloženého CD</b>	<b>46</b>
<b>B Rozšírený slovný popis 2 výsledkov Beta testov</b>	<b>47</b>

# Kapitola 1

## Úvod

Výukové nástroje rôznych foriem na výučbu hrania na hudobné nástroje sú tu s nami už stovky rokov a postupne sa rovnako, ako všetko, prispôsobujú dobe. Ešte pár desiatok rokov dozadu to boli buď klasické notové zápisy v notovej osnove, či len akási forma improvizácie a hľadania správnych tónov podľa audiozáznamov.

Dnes už nie je potrebné chodiť do hudobných škôl a učiť sa častokrát nadbytočné množstvo teoretických znalostí a venovať im veľké množstvo finančných prostriedkov a osobného času keď existujú webové aplikácie alebo online učители s kurzami, ktorí vás týmto svetom prevedú. Vo väčšine prípadov za podstatne nižší finančný obnos, ako aj vo flexibilnom čase. Veľké množstvo týchto aplikácií je ale príliš komplikovaných, najmä pre začiatočníkov odborných, neprehľadných a celkovo užívateľsky neprívetivých. Táto práca ukazuje a vysvetľuje implementáciu jednotlivých nástrojov potrebných k danej výučbe pomocou techník detekcie dominantnej frekvencie a filtrovania nežiadúcich artefaktov zvuku, či okolia.

V tomto projekte bude implementovaná aj webová aplikácia na výučbu hry na gitaru s vlastnými komponentmi a nástrojmi, ako ladička, či interaktívna tabulatúra pre jednoduchú a efektívnu výučbu hry na gitaru.

Cieľom technickej časti tejto práce je vysvetlenie fungovania jednotlivo implementovaných nástrojov vrámci webovej aplikácie a jej dopad na užívateľské rozhranie, najmä po stránke spracovania signálu a zvuku ako aj detekcie šumu a nežiadúcich vplyvov okolia pomocou algoritmu, ktorý detekuje dominantnú frekvenciu daného vstupného zvukového záznamu.

Jedným z cieľov implementácie webovej aplikácie je užívateľská prívetivosť pre rôzne skupiny hráčov na tento hudobný nástroj s primárnym zameraním na gitarových hráčov, ktorí sú začiatočníci až mierne pokročilí. Zároveň s využitím vstavovaných nástrojov je cieľom vhodná interakcia s užívateľom pre dosiahnutie očakávaných výsledkov počas výučby.

Práca je rozdelená do 9 kapitol (vrátane úvodu a záveru), od prieskumu užívateľských požiadavkov a možností konkurenčných aplikácií, cez hľadanie a experimenty vhodného algoritmu na detekciu dominantnej frekvencie, až po samotný návrh a implementáciu nástrojov pre aplikáciu s dôrazom na prívetivé užívateľské prostredie a jej následné testovanie a vývoj do budúcnosti.

Prvá časť práce sa v kapitole 2 zaoberá prieskumom cieľovej skupiny a jej požiadavkami na aplikáciu a prieskumom jednotlivých konkurenčných aplikácií aktuálne dostupných na trhu. Cieľom tejto kapitoly je sa s kombináciou týchto prieskumov dopracovať k nástrojom, ktoré by mali byť v tejto aplikácii obsiahnuté a zhodnotiť reálne možnosti implementácie týchto nástrojov.

V druhej časti tejto práce sú v kapitole 3 popísané jednotlivé metódy k dosiahnutiu čo najlepšieho výsledku v oblasti detekcie dominantnej frekvencie so zameraním na robustnosť voči prostrediu.

x V tretej časti tejto práce je v kapitole 4 popis implementácie jednotlivých nástrojov využitých vrámci aplikácie.

V štvrtej časti sa v kapitole 5 pozrieme, akými spôsobmi pristupovať k návrhu takejto aplikácie so zameraním na čo najlepšiu užívateľskú prívetivosť aplikácie (UX) pre jej desktopovú a mobilnú verziu.

V piatej časti práce sa v kapitole 6 zameriame na 2 spôsoby testovania takejto aplikácie a zhodnotíme a prakticky využijeme ich výsledky v prospech finálnej verzie aplikácie.

V poslednej časti práce sa v kapitole 7 zameriame na možnosti ďalšieho vývoja tejto webovej aplikácie a popíšeme jednotlivé spôsoby, akými budú môcť byť jednotlivé funkcionality navrhnuté a implementované.

## Kapitola 2

# Prieskum požiadavkov cieľovej skupiny a konkurenčných aplikácií

### 2.1 Súhrn metodík výučby hry na gitaru

Všeobecne môžeme rozdeliť výučbu hry na gitaru do dvoch základných kategórií, ktoré by pri výučbe gitary mali byť obsiahnuté. Jedná sa o vhodný pomer zamerania sa na teoretické základy hudobnej teórie a držania gitary pri hre v kombinácii s praktickou výukou.[12]

- **Teoretická časť** - zápis a porozumenie tradičnej notovej osnovy a tzv. tabulatúry, hudobnej harmónii, tempu a rytmike danej piesne, pochopenie ladenia jednotlivých piesní a ich hudobnej štruktúre
- **Praktická časť** - správny postup ladenia gitary a základna práca s nástrojom v zmysle údržby a spravovania nástroja, hra na gitaru so správnym prstokladom, držaním nástroja a výučba hry na gitaru podľa daného notového zápisu v správnom rytme a tempe.

Kombináciou takejto výučby s dôrazom hlavne na praktickú časť je možné u začiatočníkov dosiahnuť adekvátny progres a u pokročilejších klásť dôraz na zlepšenie jednotlivých nedostatkov či už po teoretickej, alebo praktickej stránke z ich hudobnej minulosti.

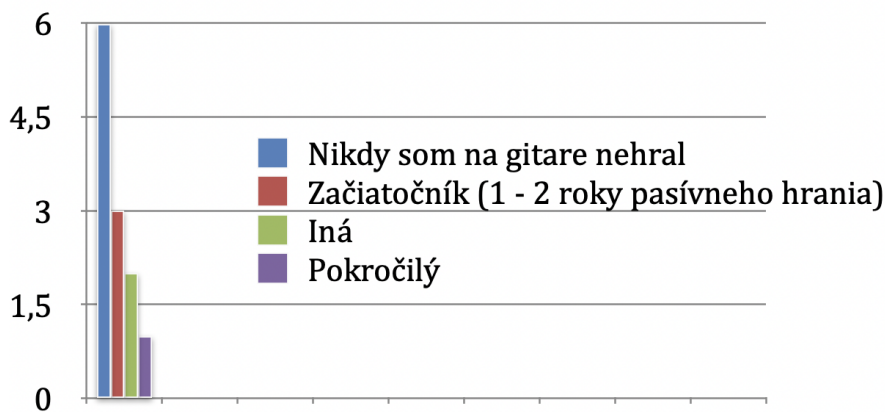
### 2.2 Cieľová skupina

Webová aplikácia pre výučbu hry na gitaru je vhodná pre gitarových hráčov rôznych úrovní. Avšak jej návrh si zakladá na prívetivom užívateľskom prostredí, ktoré dovoľuje prácu z jednotlivými nástrojmi aj pre úplných začiatočníkov bez predošlých hudobných znalostí. Taktiež na základe už naučených schopností je možný vhodný výber piesní a nastavenia vlastného tempa pre danú skladbu.

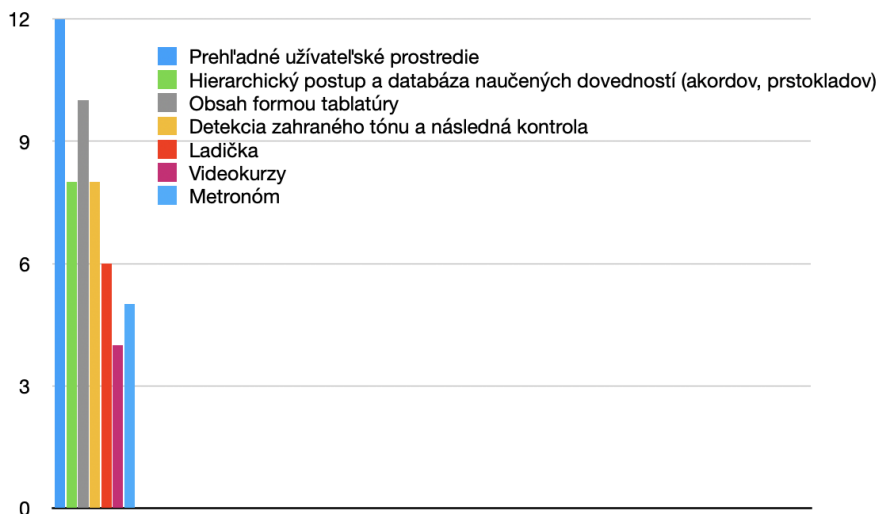
Pokročilejší hráči majú možnosť experimentovať s rýchlejšim tepom skladieb, či si vyskúšať aj technicky náročnejšie skladby, ktorými ich aplikácia prevedie a zhodnotí, či tóny v danej piesni boli odohrané korektne.

Pre bližšiu charakteristiku cieľovej skupiny bol zaslaný anonymný formulár 16 subjektom, ktorí zodpovedali na otázky z oblasti ich aktuálnej úrovne hry na gitaru a zároveň poskytli

odpovede na otázky ohľadom vstavaných nástrojov, ktoré by v danej aplikácii boli prínosné a motivovali ich k jej využitiu do budúcnosti. Formulár bol vytvorený a sprostredkovaný pomocou aplikácie Survio. Nižšie sa nachádza prehľad najdôležitejších otázok a odpovedí s krátkym popisom, ktorého výsledok mal vplyv najmä pri výbere jednotlivých nástrojov do aplikácie.



Obr. 2.1: Graf preferencií podľa skúseností opýtaných s hrou na gitaru



Obr. 2.2: Prehľad jednotlivých požiadavkov na výslednú webovú aplikáciu

Na základe výsledkov z formuláru sme sa dozvedeli, že najväčšie zastúpenie majú ľudia, ktorí na hudobný nástroj hrať nevedia ale radi by túto schopnosť získali. Začiatocníci, ako aj pokročilejší hráči by majoritne uvítali najmä prívetive a jednoduché užívateľské prostredie a vstavané nástroje, najmä ladičku a doplnkové nástroje na tréning rytmiky, tempa a detekcie jednotlivých akordov a tónov pre zdokonalenie "hudobného sluchu".

25% z opýtaných už malo predošlé skúsenosti s nejakou formou využitia webových aplikácií výučby hry na gitaru. Medzi nedostatky nimi používaných aplikácií patrili najmä nevyhovujúce a často komplikované užívateľské rozhranie aplikácie a chýbajúce alebo nedostatočné základné nástroje (ide hlavne o nástroje ladička a metronóm). Ďalším nedostatkom najmä pri open-sourcových aplikáciách bolo často veľké množstvo obsahových chýb (chybné akordy, tabulatúra a podobne).

Ďalším aspektom bola mimo open-sourcových aplikácií ich cena za mesačné predplatné na využívanie softwaru pohybujúca sa často v desiatkach eur. Majoritná časť opýtaných by privítala možnosť bezplatnej základnej časti aplikácie pri rozhodovaní sa výberu výukového programu. Konkurenčné platené aplikácie najčastejšie využívajú koncept niekoľkodňového voľného prístupu k daným funkciám aplikácie, ktorý je následne po jej ukončení spoplatnený. Táto skutočnosť má negatívny dopad na používateľov, ktorým často nestačí táto lehota na rozhodnutie sa, či danú aplikáciu majú záujem v budúcnosti využívať.

Jedna z opakovane spomínaných požiadavok bola možnosť tréningovej sekcie na tempo, rytmus. Tieto nástroje umožňujú prehĺbovať pochopenie základných hudobných prvkov, ktoré sú pri výuke nepostrádateľné.

Výsledky tohto prieskumu boli zohľadnené pri návrhu jednotlivých nástrojov a prvkov tejto aplikácie. Väčšina výsledkov je relevantná a do značnej miery splniteľná pri návrhu a implementácii tejto webovej aplikácie.

## 2.3 Rozbor aktuálne dostupných webových aplikácií výučby gitarovej hry

Webové aplikácie na výučbu gitary sú užitočným nástrojom pre hráčov, ktorí sa chcú naučiť alebo zlepšiť sa v hre na gitare. Existuje mnoho aplikácií, ktoré poskytujú rozličné lekcie a cvičenia, a výber aplikácie závisí od subjektívnych preferencií a potrieb jednotlivých hráčov. Všetky tieto aplikácie sa rozlišujú, ako náročnosťou podľa úrovne používateľa, sadou použiteľných nástrojov rámci aplikácie, tak aj a samotnou cenou (prístupnosť do týchto aplikácií je väčšinou spoplatnená vo forme mesačného alebo ročného predplatného pre využívanie danej webovej aplikácie).

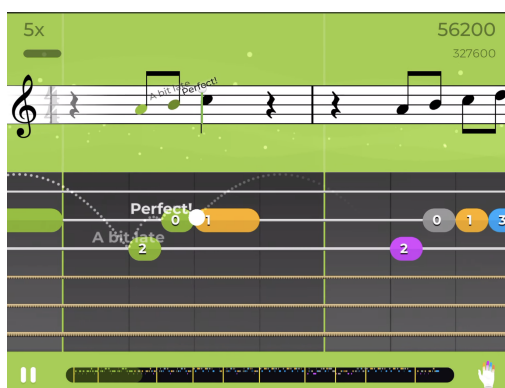
Na výskum bolo vybraných päť webových aplikácií ohodnotených, ako najlepších na aktuálnom trhu podľa stránky guitar.com[5]. Finálne výsledné hodnotenie pre jednotlivé aplikácie sú kombináciou názoru vyššie spomínaného zdroja a individuálneho testovania daných aplikácií. Toto testovanie bolo pri vybraných aplikáciách možné zdarma, vďaka ich limitovaným, bezplatným skúšobným verziám.

Za účelom testovania bola vybraná skladba Tears in Heaven, ktorú bolo potrebné v jednotlivých aplikáciách dohľadať a vyskúšať ponúkané možnosti výučby.

**Yousician**<sup>1</sup> ponúka veľmi prepracované prostredie ale pripomínajúce skôr hru, ako serióznu aplikáciu. Ide o platformu s možnosťou vlastného progresu na základe hierarchických úrovní výučbového postupu a splnenia určitých kritérií pri prechode aplikáciou.

Vzhľadom na prepracované grafické rozhranie je pomerne robustná a najmä pri jej použití na menej výkonných systémoch alebo s horším internetovým pripojením je v jej online móde pomalá a nedostatočne vyvážená.

Najväčšia výhoda tejto aplikácie (ktorou nedisponuje žiadna iná aplikácia z vybraných konkurenčných aplikácií pre tento výskum) je detekcia zahranych tónov pri hraní spolu s pohyblivou tabulatúrou. Táto funkcia je pre užívateľov jedna z najdôležitejších a veľmi dobre pomáha v celkovom zorientovaní hráča na gitarovom hmatníku pri hre na gitaru.

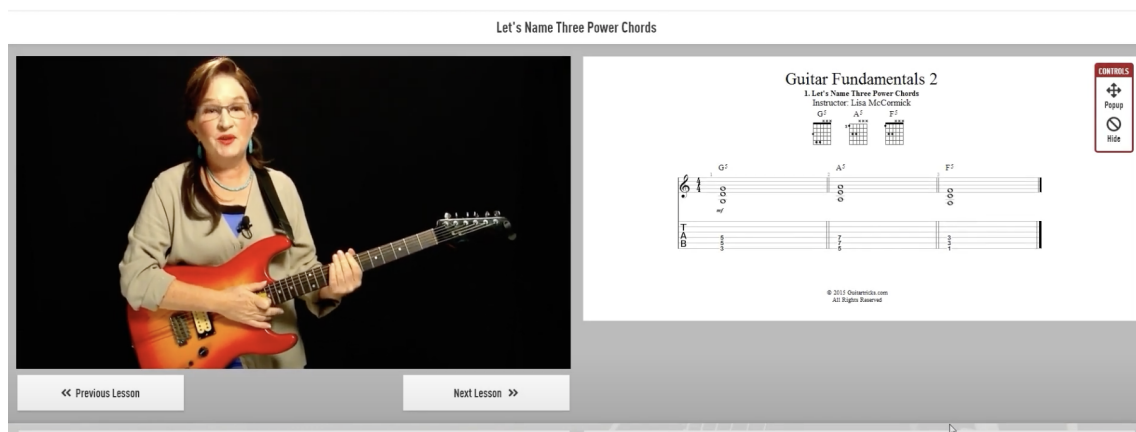


Obr. 2.3: Ukážka z experimentovania s aplikáciou Yousician

<sup>1</sup>Aplikácia Yousician je veľmi prepracovanou a kvalitnou aplikáciou s možnosťou hry pre rôzne typy gitár vrátane ukulele, či 12 - strunovej gitary

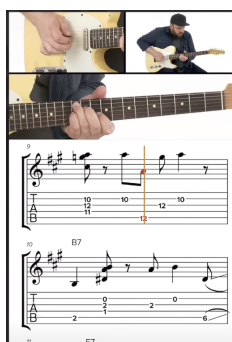
**Guitar Tricks**<sup>2</sup> ponúka obrovskú sadu videokurzov podporovaných od rôznych promovovaných hudobníkov. Zároveň je pomerne bohatá na obsah použiteľných nástrojov, ktoré ale nie sú pre začiatočníkov veľmi užívateľsky prívetivé. Výučba pomocou tejto aplikácie funguje kombináciou priloženého videokurzu a notovej osnovy vedľa videa. Vytváranie jednotlivých videokurzov pre jednotlivé skladby má nevýhodu najmä pre časovú a materiálnu náročnosť jej obsahu pre jednotlivé skladby - užívateľ tu nájde nové skladby pre výučbu až po dlhšej dobe od ich vydania.

Najväčšie mínus tejto aplikácie je cena, ktorá je najmä pre užívateľov s pomerne krátkou skúšobnou dobou na prvotné vyskúšanie webovej aplikácie nedostačujúca.



Obr. 2.4: Ukážka z webovej aplikácie Guitar Tricks

**True Fire** má k dispozícii bohatú databázu skladieb s nadštandardne prepracovanou tabulatúrou. Zároveň je doplnená videokurzami známych hudobníkov, čím sa ich kvalita do značnej miery znásobuje. Zpomedzi všetkých vyskúšaných aplikácií rámci tohto výskumu mala jednoznačne najhoršie užívateľské rozhranie aj podľa mojej recenzie, aj podľa recenzií z referenčnej stránky. V tomto prípade sa nachádza na liste aj kvôli jej cene, ktorá je vzhľadom na ponúkané služby prijateľná<sup>3</sup>.



Obr. 2.5: Ukážka z testovania tabulatúry webovej aplikácie True Fire

<sup>2</sup>Guitar Tricks je najviac promovanou aplikáciou tohto druhu medzi hudobníkmi

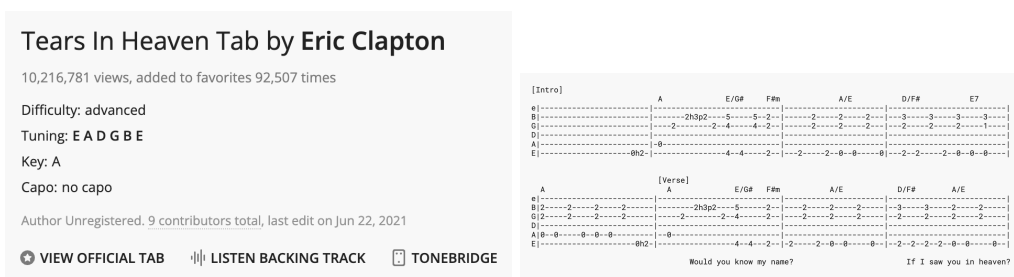
<sup>3</sup>Výhodou tejto aplikácie je najmä kvalitný obsah od tých najlepších svetových gitaristov

**Songsterr**<sup>4</sup> je výborná aplikácia, ktorú osobne využívam už približne 6 - 7 rokov. Aplikácia je open-source, takže obsah pre jednotlivé skladby do nej vkladá komunita. Tabulatúra a celkové rozhranie pôsobí dobrým dojmom s možnosťou prispôsobenia rýchlosti prehrávania a ďalšími vstavanými funkciami pre tabulatúru (tieto funkcie sú spoplatnené). Najväčšou nevýhodou sú chýbajúce základné nástroje, ako aj fakt, že obsah často nie je korektný vzhľadom na skutočnú skladbu, keďže ho spravuje pomerne široká komunita ľudí.



Obr. 2.6: Náhľad z testovania aplikácie Songsterr

**Ultimate Guitar**<sup>5</sup> má najväčšiu databázu akordov, tabov a nôt pre lokálne aj zahraničné piesne. Pri vyhľadávaní akordov k väčšine piesňam sa táto aplikácia zobrazuje v internetových prehliadačoch medzi prvými. Pridávanie obsahu funguje na obdobnom princípe, ako je to pri aplikácii Songsterr. Opäť, pridaných vstavaných nástrojov je veľmi málo a nie sú moc dobre optimalizované ani užívateľsky prívetivé. Rovnako ako pri Songsterri aj tu platí, že obsah tabov je veľmi často nesprávny, keďže je vytváraný širokou internetovou komunitou.



Obr. 2.7: Náhľadové obrázky z hry na gitaru pomocou tabulatúry aplikácie Ultimate Guitar

<sup>4</sup>Patrí medzi najväčšie zdroje tabulatúry a akordov pre široké spektrum hudobných žánrov.

<sup>5</sup>Táto webová aplikácia sa zobrazuje vždy ako prvá pri akomkoľvek hľadaní akordov alebo tabulatúry pre gitarové skladby.

Po otestovaní vyššie spomínaných webových aplikácií bola zostavená výsledna tabuľka:

Názov aplikácie	Videokurzy	Ladička	Metronóm	Tablatúra	Cena	UX	Celkové hodnotenie
Yousician	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	5 / 10	6 / 10
Guitar Tricks	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	6 / 10	8 / 10
True Fire	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3 / 10	4 / 10
Songsterr	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9 / 10	7 / 10
Ultimate Guitar	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6 / 10	7 / 10

Obr. 2.8: Prehľad jednotlivých kritérií pre konkurenčné webové aplikácie na výučbu hry na gitaru

Na záver tejto sekcie treba dodať, že všetky vyššie spomínané webové aplikácie sú korporátnych rozmerov a za ich údržbou a technickým vývojom stoja desiatky až stovky. Z tohto dôvodu **nie je možné** rámci bakalárskej práce takýmto aplikáciám priamo konkurovať. Cieľom časti práce, ktorá sa zaoberá návrhom a implementáciou vlastnej webovej aplikácie výuky hry na gitaru je vytvorenie vlastných nástrojov, vďaka ktorým sa môže určitá skupina ľudí naučiť hrať na gitare a zároveň sa nestratí vo veľmi neprehľadnom a hudobno-technicky náročnom prostredí.

## 2.4 Nástroje vyvíjanej aplikácie

Jednotlivé nástroje boli do aplikácie vybrané najmä kombináciou užívateľského dotazníka a vlastného výskumu spojeného s aktuálnou situáciou na trhu s webovými aplikáciami na učenie sa hry na gitaru. Každý nástroj má rôznu náročnosť na implementáciu po stránkach spracovania a detekcie zvuku, výpočetnej náročnosti a využívania externých knižníc a algoritmov na korektnú implementáciu.

Tu je prehľad nástrojov ktoré budú zabudované rámci tejto gitarovej aplikácie:

- **tabulatúra / notový zápis** - pre jednotlivé piesne možnosť nastavenia zobrazenia tabov, ukazovateľ na aktuálnu pozíciu v piesni a nastavenie tempa podľa potrieb užívateľa
- **Ladička** - podpora rôznych typov ladenia, možnosť vlastných metód ladenia, interval presnosti podľa výšky v Hz

Tieto nástroje budú základnou súčasťou našej webovej aplikácie. Keďže tieto nástroje nie sú úplne triviálne a ich kompozíciu, ako aj miesto v aplikácii je potrebné popísať podrobnejšie, sú popísané v ďalších kapitolách, ako aj algoritmus, ktorý využívajú k detekcii výšky tónu.

## Kapitola 3

# Konvolučné neurónové siete (CNN) a algoritmus CREPE

### 3.1 Konvolučné neurónové siete (CNN)

Konvolučné neurónové siete sú jedným z druhov hlbokých neurónových sietí. Tieto neurónové siete majú rovnako ako iné neurónové siete definované parametre (váhy), ktorých hodnoty sú predmetom dlhodobého učenia a výskumu vedeckých inštitúcií. Ich architektúru popisujeme vrstvami, ktoré obsahujú neuróny. Ich výpočet sa líši v závislosti od ich typu. Každý neurón v sieti je vypočítaný skalárnym súčinom svojho vstupu a parametrov. Na základe nelineárnej funkcie potom odchádza k aktivácii jeho výstupov. Od všeobecných neurónových sietí sa ale odlišujú najmä tým, že obsahujú tzv. konvolučnú vrstvu[10].

Konvolučné neurónové siete (CNN) majú v hudbe rôzne využitie, od rozpoznávania melódie a akordov až po celkovú kompozičnú hudobnú tvorbu.

Okrem klasifikácie žánrov sa CNN používajú aj na generovanie hudby. Generatívne modely CNN sa trénujú na rozpoznávanie vzorov a následne generujú nové vzory, ktoré sú podobné pôvodným nahrávkam. Tento prístup umožňuje vytvárať nové melódie, ktoré sa môžu použiť v rôznych hudobných projektoch.

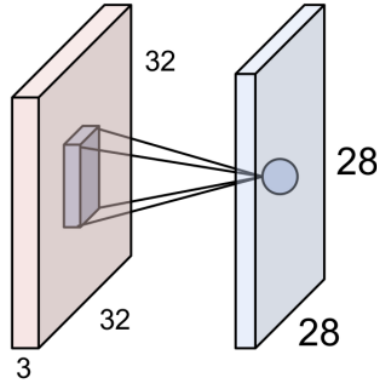
Ďalším využitím CNN v hudbe je detekcia hudobných nástrojov a zvukov. Tréningové dáta obsahujú rôzne zvukové stopy a CNN sa trénujú na rozpoznávanie týchto zvukových stôp v hudobných nahrávkach. To môže byť užitočné v rôznych aplikáciách, ako napríklad v hudobných hrách alebo softvéroch na nahrávanie hudby.[1]

Trendom posledných pár rokov je tiež využitie CNN na zlepšenie kvality zvuku v nahrávkach. Tieto modely dokážu automaticky odstraňovať šumy a nežiadúce artefakty z nahrávok a zlepšiť celkovú kvalitu zvuku.

Konvolučné neurónové siete sa od bežných neurónových sietí odlišujú tým, že obsahujú aspoň jednu vrstvu neurónov, ktorá pracuje na základe operácie konvolúcie (\*). Vstupná hodnota konvolúcie je väčšinou viacrozmerné pole hodnôt, ktoré sa nazýva tenzor. Jadro konvolúcie má zvyčajne rovnakú dimenziu ako je dimenzia vstupu. Konvolúcia v kontexte neuronových sietí je definovaná:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n)$$

Konvolúcia v neuronových sieťach prebieha tak, že filter postupne prikladáme so zvoleným krokom na vstup a vzájomne násobíme hodnoty aktuálneho prekrytia vstupu a filtra. Výsledky jednotlivých konvolúcií sa zapisujú do aktivačnej mapy. Aktivačná mapa po skončení celej operácie konvolúcie na vstupe obsahuje hodnoty skalárneho násobenia úseku vstupu a filtra po celom priestore vstupu.



Obr. 3.1: Konvolučná vrstva s jedným filtrom o veľkosti 5x5x3. Ružový hranol označuje vstup, modrý aktivačnú mapu[10].

V nasledujúcej časti tejto sekcie sa budeme venovať konkrétnym implementáciám konvolučných neurónových sietí zameraných na spracovanie zvuku. V týchto metódach je zvukový signál najskôr predspracovaný, aby sa získal spektrogram, ktorý je vizuálnou reprezentáciou frekvenčného obsahu zvukového signálu v čase. CNN potom používajú tento spektrogram ako vstup a učia sa odhadovať základnú frekvenciu ako výstup.

Výhodou použitia CNN je, že sa môžu naučiť zložité vzorce v spektrograme, ktoré zodpovedajú základnej frekvencii a ktoré nie sú ľahko viditeľné pre ľudské oko. Konvolučné vrstvy CNN vykonávajú extrakciu vlastností tým, že sa učia detekovať lokálne vzorce v spektrograme, zatiaľ čo plne prepojené vrstvy sa učia kombinovať tieto vlastnosti a produkovať konečný odhad základnej frekvencie.

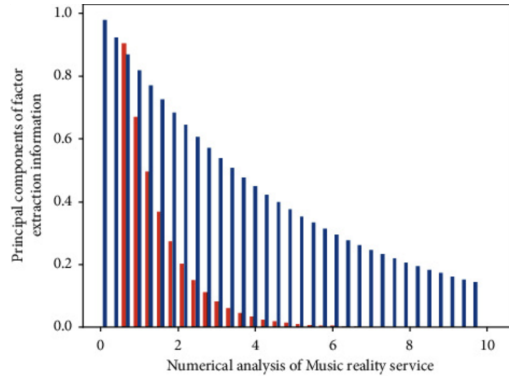
Pri výpočtoch konvolučných neurónových sietí je treba dbať na určité pravidlá a doporučenia pre optimálne výsledky z tréovania. Pri analýze jednotlivých hudobných komponentov treba dbať na zväčšovanie tréovacej vzorky. Uvedieme túto skutočnosť na príklade:

Extrahujeme náhodné  $224 \times 224$  malé časti hudobných prvkov z  $256 \times 256$  hudobných prvkov. Extrahované hudobné prvky sa líšia od pôvodných hudobných prvkov o 32 pixelov, takže hlavná časť by mala byť obsiahnutá v tréovacej sade. Jedna z metód spočíva v použití PCA (algoritmus transformácie datasetu na jednotlivé hlavné komponenty) na zlepšenie tréovacích dát: pre každý RGB hudobný prvok sa vykoná transformácia PCA na dokončenie odhlučňovacej funkcie (odstránenie nežiadúcich artefaktov zo zvukového záznamu).

$$\frac{k(m, n) - 1}{k(m, n) + 1} - \sum x(k) - \frac{k(m, n)}{k(m)k(n) - 1}$$

Pre jednotlivé analyzačné vrstvy sa zvyčajne používajú metódy priemerového delenia. Po prechode cez vrstvy majú výsledné matice polovičné hodnoty riadkov a stĺpcov oproti pred-

chádzajúcim vrstvám. Z toho vyplýva, že nastala zmena časových a frekvenčných vlastností siete, ktoré majú za následok rôzne dopady na výsledné hodnoty.



Následne je potrebné jednotlivé konvolučné vrstvy navzorkovať podľa vhodnej vzorkovacej frekvencie.

Pre maximálnu frekvenciu, ktorá môže byť zaznamenaná (napr. 20000 Hz), je potrebných aspoň 40 000 vzoriek za sekundu. Zvyčajná vzorkovacia frekvencia pre reálne audio súbory je 44 100 vzoriek za sekundu, čo je o niečo viac, ako žiadúcich 40 000, takže musíme pri kompresii brať do úvahy túto skutočnosť a použiť aspoň 40 000 jednotiek na vstup. Treba brať do úvahy jednotlivé váhové hodnoty a podľa nich vhodne prispôbovať dáta.

## 3.2 CREPE

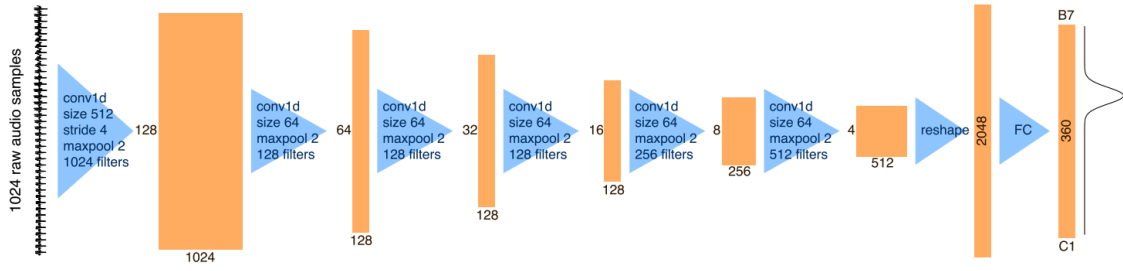
Úlohou odhadu základnej frekvencie monofónneho nahrávania zvuku. Monofónny zvuk je zvukový systém, v ktorom sú zvukové signály zmiešané a potom smerované cez jeden zvukový kanál[17]. Je využívaná pri spracovaní zvuku mnohými aplikáciami v spracovaní reči a získavaní potrebných informácií zo zvukovej stopy. Doposiaľ jeden z najpoužívanejších algoritmov pre túto úlohu je pYIN[11]. Slúži na odhadovanie základnej frekvencie na ktoré využíva metódu prahového delenia. Aj keď takáto funguje relatívne správne, existuje mnoho prípadov, kedy dôjde nedostatkom tejto metódy k nesprávnym odhadom, ktoré majú za následok isté percentuálne výchylky. Aj z tohto dôvodu preto bol pre účely našej aplikácie zvolený iný algoritmus z prostredia konvolučných neurónových sietí.

Algoritmus založený na detekcii dominantnej frekvencie - CREPE je založený na hlbokéj konvolučnej neurónovej sieti, ktorá priamo pracuje na časovej doméne vstupného signálu. Odhad základnej frekvencie monofónneho zvukového signálu (odhad frekvenčnej výšky tónu), je dlhodobou výskumnou témou v spracovaní zvukového signálu.

Výška tónu sa definuje kvalitou vnímaných hudobných tónov a presne nezodpovedá fyzikálnym a matematickým vlastnostiam základnej frekvencie, avšak pre majoritnú časť prípadov je možné výšku tónu kvantifikovať práve pomocou základnej frekvencie daného zvukového záznamu.

### 3.2.1 Architektúra algoritmu CREPE

CREPE sa skladá z hlbokého konvolučnej neurónovej siete, ktorá pracuje so zvukovým signálom na jej časovej doméne a produkuje odhad výšky tónu.



Obr. 3.2: Architektúra algoritmu CREPE

Na začiatku prehľadu architektúry CREPE sú vstupné dáta pozostávajúce z 1024 vzorkov z audiosignálovej časovej domény s použitím vzorkovacej frekvencie 16 kHz. Nachádza sa tu šesť konvolučných vrstiev, ktorých výsledkom je 2048 rozmerná latentná reprezentácia, ktorá je napojená na výstupnú vrstvu zodpovedajúca 360 rozmernému výstupnému vektoru  $\hat{y}$ . Výsledný odhad výšky tónu je na základe tohto výstupného vektoru vypočítaný deterministicky.

Každý z 360 uzlov vo výstupnej vrstve zodpovedá špecifickej hodnote výšky tónu, definovanej v centoch. Cent je bezrozmerná fyzikálna jednotka pre meranie veľkosti intervalov, používaná v hudobnej akustike a hudbe všeobecne vzhľadom na referenčnú výšku tónu frekvencie v Hz:

$$c(f) = 1200 \cdot \log_2 \frac{f}{f_{ref}}$$

kde sa počas príslušných experimentov používa  $f_{ref} = 10$  Hz. Táto jednotka poskytuje logaritmický rozsah výšky tónu, kde 100 centov sa rovná jednej polovici tónu. Hodnoty výšky tónu 360 sú označené ako  $\hat{c}_1, \hat{c}_2, \dots, \hat{c}_{360}$  a sú vybrané tak, aby pokryli šesť oktáv s 20-centovými intervalmi medzi C1 - B7, čo zodpovedá 32,70 Hz a 1975,5 Hz. Výsledný odhad výšky tónu  $12\hat{c}$  je vážený priemer súvisiacich výšok  $c_i$  podľa výstupu  $\hat{y}$ , ktorý udáva odhad frekvencie v Hz:

$$\hat{c} = \frac{\sum_{i=1}^{360} \hat{y}_i c_i}{\sum_{i=1}^{360} \hat{y}_i}, \hat{f} = f_{ref} \cdot 2^{\frac{\hat{c}}{1200}}$$

Cieľové výstupy, ktoré boli používané vrámci experimentov referenčnej práce na trénovanie modelu, sú 360-rozmerné vektory, kde každý rozmer predstavuje frekvenčný zásobník pokrývajúci rozsah 20 centov (rovnako ako výstup modelu).[2]

$$y_i = \exp\left(-\frac{(c_i - c_{true})^2}{2 \cdot 25^2}\right)$$

Týmto spôsobom vysoké aktivačné hodnoty v poslednej vrstve naznačujú, že vstupný signál bude mať pravdepodobne výšku tónu, ktorá je blízka pridruženým výškam tónov uzlov s vysokými aktivačnými hodnotami. Trénovaná tak, aby minimalizovala binárnu krížovú entropiu medzi cieľovým vektorom  $y$  a predpovedaným vektorom  $\hat{y}$ :

$$s(y, \hat{y}) = \sum_{i=1}^{360} (y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i))$$

kde  $y_i$  aj  $\hat{y}$  sú reálne čísla medzi 0 a 1. Táto funkcia stratovosti je optimalizovaná pomocou optimalizátora ADAM<sup>1</sup> s rýchlosťou učenia 0,0002. Najvýkonnejší model sa vyberie po

<sup>1</sup>je stochastický algoritmus pre optimalizáciu stochastických objektovo orientovaných funkcií založený na adaptívnych odhadoch momentov nižšieho stupňa.

tréningu v prípade že sa daná trieda už nezlepšuje. Každý konvolučnej vrstve predchádza dávková normalizácia a po nej nasleduje vrstva vypadnutia s pravdepodobnosťou vypadnutia 0,25. Táto architektúra a tréningové procedúry sú implementované s využitím python knižnice Keras obsahujúcej vstavané funkcie pre hlbkové učenie.

Keras je API využívajúce hlboké učenie. Bola vytvorená spoločnosťou Google na zjednotenie implementácie neurónových sietí [2].

### 3.3 pYIN a SWIPE

PYIN algoritmus používa pravdepodobnostné prahové rozdelenie na modelovanie prítomnosti alebo neprítomnosti parciálnych frekvencií pri každej harmonické frekvencii. Práve toto prahové rozdelenie sa používa na oddelenie harmonických kmitočtových vrcholov od šumu a iných rušivých elementov v zvukovom signáli.

V rámci článku je vykonané hodnotenie výkonu PYIN algoritmu na základe rôznych zvukových záznamov, vrátane pomerne zaujímavého zloženia datasetu obsahujúceho hranie sólového hráča, ako aj hru zborového orchestru. Výsledky ukazujú, že algoritmus dobre odhaduje základné frekvencie aj v prítomnosti šumu a prekrývajúcich sa harmonických kmitočtov.

[11] Celkovo článok ukazuje, že PYIN algoritmus je efektívna a spoľahlivá metóda na odhad základných frekvencií v hudobných signáloch, s potenciálnym využitím v automatickej transkripcii a získavaní informácií o hudbe.

SWIPE je jeden z ďalších a zároveň najstarším algoritmom spomínaným v tejto práci na odhadovanie výšky tónu zvukových signálov. Algoritmus vychádza zo základného princípu, že prirodzené zdroje zvuku ako hlas alebo hudobné nástroje produkujú zvukové signály s istou periodickou štruktúrou. Z tohto dôvodu algoritmus SWIPE využíva na odhad výšky tónu jednotlivé dáta o pravidelnosti periodických signálov.

Algoritmus SWIPE je inšpirovaný takým signálom, ktorý má periodickú a jednoduchú štruktúru. Algoritmus využíva koreláciu medzi pôvodným signálom a tzv. pilovým signálom na výpočet frekvencie. V prípade, že vstupný signál nie je úplne periodický, algoritmus SWIPE vie využiť rôzne ďalšie techniky pre nájdenie a výpočet výšky jeho tónu.

V rámci článku z ktorého boli čerpané informácie pre algoritmus SWIPE sú prezentované výsledky porovnávajúce SWIPE algoritmus s existujúcimi algoritmi na odhadovanie tónovej výšky. Výsledky ukazujú, že algoritmus SWIPE dosahuje lepšie výsledky ako väčšina existujúcich algoritmov, a to aj pri použití nízko-kvalitných zvukových záznamov a pri odhadovaní harmonických zložiek v hudobných signáloch.

Celkovo článok ukazuje, že SWIPE algoritmus je účinným a spoľahlivým spôsobom na odhadovanie tónovej výšky v zvukových signáloch, s potenciálom využitia v oblasti automatickej transkripcie a získavania informácií o hudbe.

V nasledujúcich sekciách využijeme informácie o všetkých 3 algoritmoch a z vedeckej práce o CREPE[2] použijeme ich experimentálnu časť v ktorej je práve CREPE porovnávaný s algoritmi pYIN a SWIPE.

### 3.4 Experimentálna časť

Táto sekcia nie je experiment s vlastnými dátami ale vychádza z referenčnej práce CREPE[2]. Hlavným dôvodom, prečo nebolo možné vytvoriť vlastný experiment na rozumnej úrovni, bola prílišná náročnosť datasetu používaného v experimente zo zdroja, ktorý bol jedným z tých najdôležitejších pri príprave tejto práce.

Aby bolo možné jednoznačne zhodnotiť technické možnosti algoritmu CREPE a porovnať jeho výkon s alternatívnymi algoritmi na báze konvolučnej neurónovej siete, je potrebné mať k dispozícii dostatočnú vzorku experimentálnych dát. Aby bolo zaručené objektívne hodnotenie, bude potrebné použiť datasety syntetizovaných zvukov, kde je v plnej réžii možnosť kontroly základná frekvencia výsledného signálu. Boli použité tieto 2 datasety:

- RWC-synth, ktorý obsahuje 6,16 hodiny zvukového záznamu syntetizovaného z RWC Music Database a bol použitý na porovnanie s predošlým algoritmom pYIN a algoritmom SWIPE. Ide o zjednodušený scenár oproti skutočnosti
- dataset s obsahom 230 monofónnych zvukových záznamov získaných z MedleyDB a syntetizovaných pomocou metodológie, ktorá využíva analýzu na vytvorenie syntetizovaného záznam, ktorá zachováva zvukový prejav a dynamiku pôvodných zvukových stôp

Dataset	Metric	CREPE	pYIN	SWIPE
RWCsynth	RPA	0.999±0.002	0.990±0.006	0.963±0.023
RWCsynth	RCA	0.999±0.002	0.990 ±0.006	0.966±0.020
MDBstems	RPA	0.967 ±0.091	0.919 ±0.129	0.925±0.116
MDBstems	RCA	0.970±0.084	0.936 ±0.092	0.936±0.100

Obr. 3.3: Výsledky experimentu v závislosti od použitého datasetu. RPA (Raw Pitch Accuracy), RCA (Raw Chroma Accuracy)

Na základe vyššie zobrazenej tabuľky výsledkov experimentov s použitím 2 rozdielnych datasetov môžeme skonštatovať, že bez ohľadu na dataset a použitú metriku vrámci daného datasetu algoritmus CREPE mal v každom ohľade lepšie celkové hodnotenie, ako algoritmy pYIN a SWIPE.

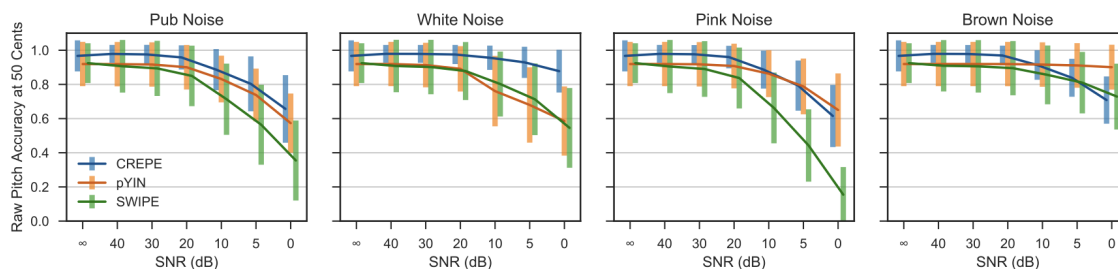
### 3.4.1 Výsledky vzhľadom na detekciu zahraniého tónu

Tabuľka 1 ukazuje výkon odhadu výšky tónu testovaný na dvoch datasetoch. Na datasete RWC-synth dosahuje CREPE takmer 100 % výsledky narozdiel od konkurenčných algoritmov, kde chybovosť je o viac ako 10% nižšia ako u ostatných algoritmov. Tieto dáta ale sú mierne skreslené vzhľadom na použitý dataset.

Vzhľadom na potrebu dosiahnutia aj praktických a nie len teoretických výsledkov, bol tiež vyhodnotený výkon na datasete MDB-stem-synth. Napriek tomu, že vzhľadom na vstupné dáta sú výsledky menej priaznivé ako pri prvom datasete, sú stále podstatne pozitívnejšie narozdiel od konkurenčných algoritmov.

### 3.4.2 Výsledky vzhľadom na akustickú robustnosť prostredia

Odolnosť voči nežiadúcim artefaktom vyvolávané okolitým prostredím je pre našu aplikáciu veľmi dôležitá, keďže chceme zamedziť ostatným zvukovým vlnám na frekvenciách odlišných od tej dominantnej a nami očakávanej. Na obrázkoch nižšie môžeme vidieť prehľad jednotlivých algoritmov zapojených do experimentu a vplyv rôznych druhov nežiadúcich signálov.



Obr. 3.4: Sledovanie výšky tónu s prítomnosťou aditívneho šumu. CREPE preukazuje vo všeobecnosti najvyššiu odolnosť proti šumu. Jedinou výnimkou je reakcia na hnedý šum

Z tohto experimentu teda preukázateľne vyplýva, že CREPE je pre našu aplikáciu najvhodnejšia voľba, a teda bude využitý aj v rámci našej webovej aplikácie výučby hry na gitaru.

## Kapitola 4

# Nástroje aplikácie

### 4.1 Výber použitých implementačných prostriedkov

Záznam gitarových tónov je nutné snímať s čo najkvalitnejším vstupným mikrofónom. Práca so zvukovým záznamom a zaznamenávaním jednotlivých tónov je spracovávaná a implementovaná pomocou open-source javascriptových knižníc `ml5.js`<sup>1</sup>. Frontendová časť je implementovaná pomocou javascriptovej knižnice `jQuery`, ktorá je pre našu aplikáciu najvhodnejšia kvôli širokej škále možností práce s jej vstavanými funkciami a zároveň jej optimalizáciou a dokumentačnou prehľadnosťou.

Ďalšími nástrojmi na úpravu vizuálnej časti aplikácie sú kaskádové štýly (CSS), ktoré sú v rámci aplikácie dopĺňované vstavanými prvkami frameworku `Bootstrap`. Jednotlivý výber, návrh a implementácia nástrojov vychádza z týchto hlavných bodov, ktoré boli skúmané v tejto práci:

- Výstup z formuláru od potencionálnych užívateľov z prvej kapitoly
- Výber a experimenty s cieľom výberu vhodného algoritmu pre detekciu dominantnej frekvencie zo zvukového záznamu
- Časové a odborné znalosti z oboru hudobnej teórie a implementačné znalosti na vývoj daných vstavaných nástrojov pre aplikáciu
- Výstupné dáta z alfa a beta testov a implementácia týchto skutočností na dosiahnutie, čo najlepšieho celkového záveru

### 4.2 Knižnica `ml5.js` a jej využitie v aplikácii

`ML5.js` je JavaScriptová knižnica pre strojové učenie, ktorá umožňuje vývojárom vytvárať interaktívne aplikácie a webové stránky s využitím techník strojového učenia a umelej inteligencie. `ML5.js` je postavená na populárnej knižnici `TensorFlow` a umožňuje vývojárom vytvárať a trénovať rôzne modely strojového učenia, ako napríklad klasifikátory obrazov a zvukov, generátory textu či obrazov.

`ML5.js` je navrhnutá pre použitie v prehliadači a umožňuje vývojárom vytvárať aplikácie v JavaScripte pomocou jednoduchého API.

---

<sup>1</sup>`ml5.js` slúži na jednoduché praktické využitie strojové učenia širokej škále programátorov a študentov. Knižnica poskytuje prístup k algoritmom a modelom strojového učenia v prehliadači, pričom je postavená na knižnici `TensorFlow.js` a jej využitie je možné bez potreby dodatočných inštalácií[13].

Algoritmus detekcie výšky tónu je spôsob odhadu výšky tónu alebo základnej frekvencie zvukového signálu. Táto metóda umožňuje použiť predtrénovaný model strojového učenia na detekciu výšky tónu zvukového súboru. Vrámci funkcií tejto knižnice na detekciu výšky tónu je využívaný algoritmus CREPE, ktorý bol podrobne popísaný v kapitole 3.

Najdôležitejšie funkcie využité aj vrámci našej aplikácie sú:

- funkcia `setup` - vrámci tejto funkcie sú volané metódy `getAudioContext()`, ktorá vracia objektovú verziu odchyteného tónu a `AudioIn()` uloží záznam o aktuálne využívanom vstupnom mikrofóne
- funkcia `listening` - vrámci funkcie je využívaná metóda `pitchDetection(model_url, audioContext, mic.stream, modelLoaded)`, kde `model_url` je odkaz na modely repozitáru algoritmu CREPE a `mic.stream` je záznam zo vstupného zariadenia na meranie vstupného zvukového signálu
- funkcia `getPitch(frequency)` - pri zavolaní tejto funkcie je detekovaná frekvencia vo výške `frequency` a jej následná práca aplikáciou[9]

Konkrétna implementácia vrámci našej aplikácie v súbore `tuner.js` vyzerá nasledovne:

```
const model_url = "path/to/crepe/models";
let mic;

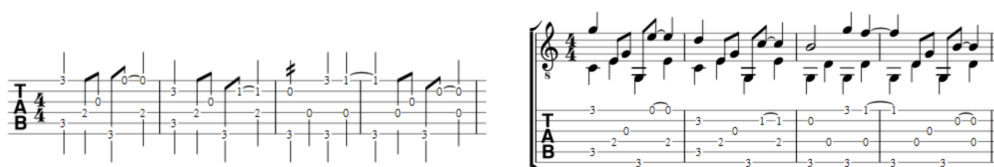
function setup() {
  audioContext = getAudioContext();
  mic = new p5.AudioIn();
  mic.start(listening);
}

function listening() {
  console.log("listening");
  pitch = ml5.pitchDetection(model_url, audioContext, mic.stream);
}

function getPitch(error, frequency) {
  if (error) {
    console.error(error);
  } else {
    if (frequency) {
      freq = frequency;
      console.log(frequency);
    }
  }
}
```

### 4.3 Tabulatúra v reálnom čase a jej funkcie

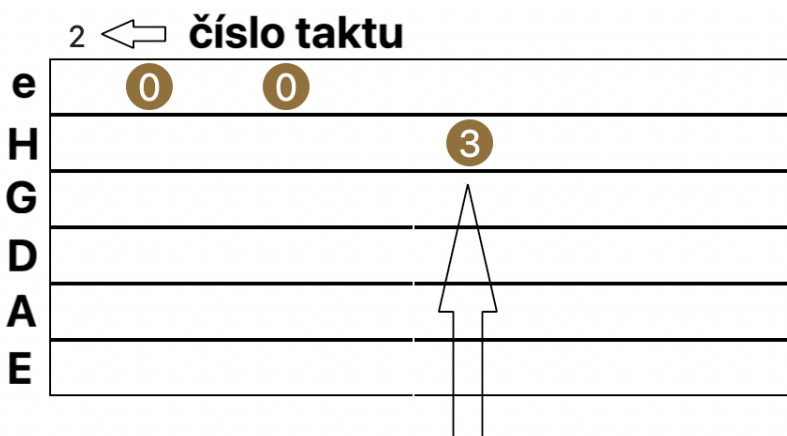
Hudobný zápis pre strunové a pražcové hudobné nástroje sa zvyčajne zapisuje v tabulatúrach (skrátene tab), ktoré sú istým typom vizualizácie strún a čísel pražcov. Tabulatúra bola bežná v období renesancie a baroka a dnes sa bežne používa pri označovaní mnohých foriem hudby vrátane zápisu gitarovej hry. Zatiaľ čo štandardná notácia predstavuje rytmus a trvanie každej noty a jej výšku vzhľadom na stupnicu založenú na dvanásťtónovom rozdelení oktávy, tabulatúra je namiesto toho založená na metódach určujúcich kde a kedy by sa mal umiestniť prst na rozoznenie danej struny, tj. výška tónu je vizualizovaná v implicitnej forme. [21]



Obr. 4.1: Varianty prevedenia zrozumiteľnej formy tabulatúry pre gitarové nástroje

Pri návrhu tabulatúry v tomto projekte boli využité kombinácie spôsobov tabulatúry aj na základe metód zobrazených na obrázkoch vyššie. Tabulatúra v tejto aplikácii kladie dôraz na čo najjednoduchšie a najzrozumiteľnejšie podanie hudobného zápisu aby bola pochopiteľná aj pre začiatočníkov.

Vodorovné čiary určujú strunu, najhrubšia struna „E” je zobrazená v dolnom rozmedzí poslednej a predposlednej vodorovnej čiari, najtenšia struna „e” udáva rozmedzie prvých dvoch vodorovných čiar. Čísla zobrazujú pozíciu prstu na hmatníku, resp. pražec. [14] Pre



Obr. 4.2: Príklad zobrazenia jedného taktu vrámci tabulatúry s popisom jednotlivých, reprezentovaných gitarových strún

jednoznačné pochopenie zobrazených čísel v jednotlivých taboch je po prejdení kurzorom myši (desktopová verzia) alebo kliknutí na daný tab (mobilná verzia) zobrazená nápoveda, predstavujúca konkrétnu pozíciu tónu na hmatníku, konkrétnej strune a pražci, ktorú je potrebné aby užívateľ zahral.

Ďalšou dôležitou súčasťou pri návrhu zrozumiteľnej tabulatúry je správna forma vykres-

🎵	1	2	3	4	5	6	7	8	9	10	11	12
e												
H			D									
G												
D												
A												
E												

Obr. 4.3: Nápoveda zobraziteľná v rámci tabulatúry pre každú skladbu v aplikácii

lenia dĺžky zahraného tónu. Keďže táto tabulatúra nevyužíva spôsoby vykreslenia, ako pri tradičnom notov zápise je dĺžku zahraného tónu možné vyčítať pomocou jednotlivých vzdialeností medzi tónmi v jednom takte.

Výpočet vzhľadom na dĺžku jedného zobrazeného taktu prebiehal nasledovne:

Nech je dĺžka jedného zobrazeného taktu 400px. Vzhľadom na túto hodnotu boli definované 4 premenné:

- **notesMover** - vykresľovacia premenná, ktorá sa mení absolútne v rámci daného zobrazeného taktu
- **totalDuration** - Celková dĺžka jednotlivých nôt pre daný takt (pri 4/4 takte je očakávaná celková dĺžka nôt 4)
- **totalStringNotes** - Celkový počet nôt obsiahnutých v jednom takte pre danú strunu (4/4 tak môže obsahovať napríklad 4 štvorštvrtové noty alebo jednu polovú notu a dve štvorštvrtové noty)
- **notesPerTab** - Celkový počet nôt obsiahnutých v jednom takte

Aplikácia využíva jednoduchý cyklus v ktorom iteruje dáta pre každú notu danej skladby. Jedna nota je v rámci aplikácie definovaná objektom s parametrami, ako pozícia na hmatníku (tj. struna a pražec), dĺžka noty a výška jej frekvencie v Hz.

Takéto objekty sú následne uložené do poľa pomocou funkcie `Note(string notePosition, integer noteDuration, string songName)` tj. napríklad

`Note("G3", 1, "Twinkle Twinkle Little Star")`. Tento zápis hovorí, že definujeme notu na pozícii struny G za 3. gitarovým pražcom pre skladbu Twinkle Twinkle Little Star. Jednotlivá postupnosť nôt v skladbe je už určená poradím volania tejto funkcie pre danú skladbu.

Keďže tabulatúra, ako taká je len akýmsi zobrazením notového zápisu, vrámci aplikácie sú k nej pridané funkcie s pomocou ktorých je vrámci aplikácie pre užívateľa interaktívna. Tu je prehľad jednotlivých funkcií implementovaných pre tabulatúru:

- **bpmRange** - Zmena rýchlosti úderov za minútu pre danú skladbu
- **Replay song** - Spustí proces prehrávania danej skladby, počas ktorej sú prehrané jednotlivé noty na základe ich výšok v Hz
- **Start** - Spustí proces, pri ktorom sa na základe nastavenej rýchlosti danej skladby detekuje zahranie jednotlivých tónov po celú dobu jej trvania
- **Pause** - Zastavenie prehrávania alebo hrania skladby
- **Stop** - Počas prehrávania alebo hrania skladby sa daný proces zastaví a presunie sa úplne na začiatok skladby

Aby mohli tieto funkcie vrámci aplikácie pracovať správne, je nutná korektná práca s jednotlivými bežiacimi procesmi aplikácie. JavaScript sa líši od jazykov ako Java alebo C++ tým, že je jednovláknový a sekvenčný, a preto využíva tzv. asynchrónne spracovanie, ktoré neblokujú ďalší tok programu.

Pri asynchrónnom spracovaní je zvyčajne potrebné reagovať na dokončenie asynchrónneho spracovania (napríklad pokračovať v práci s jeho výsledkom) a k tomu používa JavaScript tzv. callback funkcie. Tento princíp je založený na funkcionálnom programovaní, kde funkcia môže byť premennou a tak môže byť použitá ako parameter alebo návratová hodnota inej funkcie[16].

Typickým príkladom takého asynchrónneho spracovania sú udalosti tlačidiel. Callback funkcia sa vykoná, ako reakcia v momente udalosti (click, mouseenter, mouseleave a pod.). Callback funkcia sa teda nevyhodnotí ihneď a zatiaľ čo sa nevyhodnocuje, neblokuje ďalšie spracovanie hlavného toku programu (takéto callback funkcia využíva aj nami využívaná knižnica jQuery).

V JavaScripte sú asynchrónne I/O operácie, ako AJAX, Web Sockety, databázové dotazy a podobné.

Pre potreby našej aplikácie ale nebudú dostačujúce len vstavané asynchrónne funkcie knižnice jQuery.

V nasledujúcej ukážke je časť kódu našej implementácie využívajúca asynchrónnú funkciu `run`, ktorá je využívaná pre korektnú prácu tabulatúrových funkcionalít `Pause` a `Stop`:

```
const sleep = (ms) => new Promise((r) => setTimeout(r, ms));
const elPause = document.querySelector(".js-pause");
const elStop = document.querySelector(".js-stop");

let pause = Promise.resolve();
let stop = Promise.resolve();

const pauseClick = (cb) =>
  elPause.addEventListener("click", cb, { once: true });

const stopClick = (cb) =>
  elStop.addEventListener("click", cb, { once: true });
```

```

function checkForPause() {
  pauseClick(() => {
    pause = new Promise((resolve) => {
      pauseClick(() => {
        resolve();
        checkForPause();
      });
    });
  });
}

```

```

function checkForStop() {
  stopClick(() => {
    stop = new Promise((resolve) => {
      stopClick(() => {
        resolve();
        checkForStop();
        return;
      });
    });
  });
}

```

```

checkForPause();
checkForStop();

```

```

async function run() {

  await sleep(xxx);
  await pause;
  if (await stop) {
    return;
  }
}

```

```

run();

```

V `async function run` je zavolaná funkcia `sleep` vždy po dobu prechodu na ďalší tón vypočítaná s funkcionalitou `bpmRange` vrámci ktorej je nastavené aktuálne tempo skladby. Pozastavenie touto funkciou je spôsobené vstavanou asynchronou funkciou JavaScriptu `setTimeout(callback, sleeptime(ms))`, ktorá pozastaví proces po dobu `sleeptime` udanou v milisekundách. Pokiaľ nie je stlačené tlačidlo `Pause` alebo `Stop` prebieha v tejto funkcii cyklus `for`.

V prípade, že užívateľ stlačí tlačidlo `Pause`, je vo funkcii `checkForPause()` detekovaná táto udalosť, ktorá čaká na opätovné stlačenie tohto tlačidla pre pokračovanie pôvodného procesu.

V prípade, že užívateľ stlačí tlačidlo `Stop` bude celý proces ukončený a pomocou návratového príkazu `return` tento proces ukončený.

Funkcie `Pause` a `Stop` sú využívané pri funkcionalitách `Replay Song` aj `Play`. Hlavný rozdiel medzi týmito dvoma funkcionalitami udáva využitie funkcií `play()` a `getPitch()`.

Funkcionalita `Replay song` využíva rozhranie `OscillatorNode`. Reprezentuje periodickú vlnovú funkciu, ako napríklad sínus. Je to modul na spracovanie zvuku, ktorý spôsobí vytvorenie zadaného frekvenčného tónu v danom signále, vytvorí konštantný tón v istej frekvenčnej výške[3].

Nasledujúci príklad demonštruje nami vytvorenú funkciu `play` a ukážku jej využitia vrámci kódu pri stlačení funkcionality `Replay Song`:

```
const play = (frequency = 300, duration = 1e3) => {
  const context = new AudioContext();
  const gainNode = context.createGain();
  const oscillator = context.createOscillator();
  oscillator.frequency.value = frequency;
  oscillator.connect(gainNode);
  gainNode.connect(context.destination);
  oscillator.start(0);
  setTimeout(() => oscillator.stop(), duration);
};
play(hz, duration);
```

Vďaka tejto funkcií je možné vrámci cyklického prechádzania jednotlivými tónmi skladby daný tón prehrať v jeho frekvenčnej výške po určitú dobu.

Funkcionalita `Play` naopak nepotrebuje daný tón prehrať ale zistiť, či tón očakávaný v istej frekvenčnej výške bol naozaj zahráný užívateľom na gitare. Toto je schopná detekovať funkciou, ktorá je zabezpečená knižnicou `ml5.js`, konkrétne funkciou `getPitch(hz)`, kde `hz` je očakávaná frekvenčná výška tónu.

Vzhľadom na rôzne typy gitár a finálneho naladenia gitary vznikajú isté odchýlky. Táto skutočnosť bola zohľadnená pri detekcii výšky tónu, a síce formou definovania určitej odchýlky, vrámci ktorej je tón aplikáciou akceptovaný v rozmedzí  $\pm 5Hz$  od očakávanej frekvenčnej výšky tónu.

Taktiež sa očakáva určitá časová odchýlka pri zahrnutí tónu užívateľom spôsobená jeho reakčným časom. Keďže priemerný reakčný čas človeka je asi 200 ms[19], bola táto skutočnosť tiež zohľadnená pri implementácii a pri zahrnutí daného tónu bola určená odchýlka  $\pm 100ms$  od reálne očakávaného času zahrnutia tónu aplikáciou.

## 4.4 Gitarová ladička

Ladička je typ hudobného zariadenia, ktorá buď vydáva tón s presnou frekvenciou alebo má schopnosť určiť frekvenciu znejúceho tónu. Jej najčastejšie využitie je práve pri ladení hudobných nástrojov podľa určitého typu ladenia.

V minulosti, pred rozvojom internetu, alebo priamo na koncertných pódioch sa využívajú elektronické ladičky. Merajú frekvenciu vstupného tónu a detekujú jeho absolútnu výšku a jeho rozdiel oproti referenčnému tónu. Väčšinou sú vybavené ručičkovým indikátorom alebo stupnicou zloženou z svetelných diód alebo tekutého kryštálu. Moderné elektronické ladičky umožňujú individuálnu kalibráciu a majú veľký rozsah detekovateľných frekvenčných výšok.

S pribúdajúcimi možnosťami webových prehliadačov a softvérov na detekciu frekvenčných výšok z mobilných a externých mikrofónov sa stali populárne aj ladičky dostupné v rámci webových aplikácií. Tieto ladičky fungujú na podobnom princípe ako tie fyzické avšak bez hardvérových potrieb.

Základnou funkciou gitarovej ladičky je zmena ladenia pre jednotlivé skladby. Poznáme



Obr. 4.4: Prehľad aktuálne dostupných zariadení slúžiacich na ladenie gitary

niekoľko typov ladenia pre klasickú gitaru, avšak najčastejšie ladenie je tzv. **Štandardné ladenie (E)**, kde sú jednotlivé struny (od najhrubšej po najtenšiu) ladené v poradí E, A, D, G, H, e.

V nasledujúcej tabuľke môžeme vidieť aj ďalšie využívané druhy ladenia klasickej gitary, ktoré sú implementované aj pre ladičku v našej aplikácii: Keďže ladičky na gitaru v online

Tuning type	String tuning
Standard Tuning	E-A-D-G-B-e
Open D	D-A-D-F#-A-D
Drop D	D-A-D-G-B-e
Drop 1/2	D#, G#, C#, F#, A#, D#

Obr. 4.5: Prehľad najčastejšie používaných typov ladenia pre klasickú (akustickú) gitaru

priestore nie sú v dnešnej dobe ničím výnimočným môže byť implementácia v rámci našej webovej aplikácie považovaná za zbytočnú. V tomto prípade ale má takto vstavaná funkcia ladičky 2 výhody:

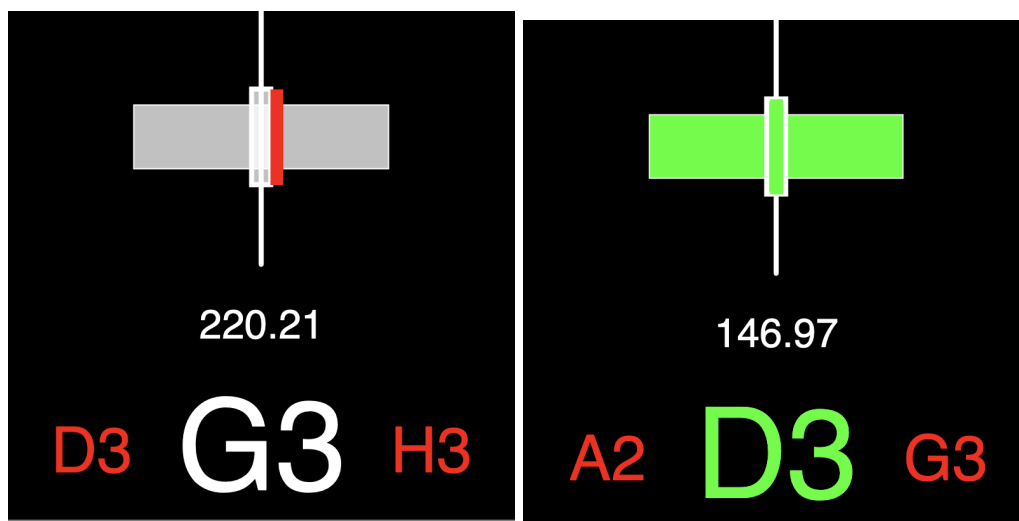
- Užívateľ ju má priamo k dispozícii po vybratí skladby automaticky alebo si môže opätovne svoju gitaru naladiť kedykoľvek počas hry
- Ladenie gitary sa automaticky zmení pre ladičku podľa daného ladenia skladby

Implementácia ladičky pre našu aplikáciu využíva funkcie z knižnice ml5.js. Okrem funkcií na nastavenie počúvania mikrofónu z predchádzajúcej sekcie využíva (rovno ako pri

počúvaní daného tónu pri tabulatúre) funkciu `getPitch(hz)`. Narozdiel od prístupu v tabulatúre je ale táto funkcia opakovane volaná formou `callbacku` dokým užívateľ neklikne mimo vykresleného prostredia pre ladičku (je zobrazená v tzv. `modal` okne).

Táto skutočnosť je detekovaná pomocou premennej typu `boolean` `isTuning`, ktorá volá dokola funkciu `getPitch` dokým `isTuning = true`.

Samotný vzhľad ladičky bol navrhnutý a implementovaný v prostredí `canvas`. Priestor, rámci ktorého je ladička vykresľovaná je automaticky daný už vo funkcii `setup` kde je vykreslená pomocou funkcie `createCanvas(400, 400)`. Vykreslenie ladičky na obrazovku je dané pomocou funkcie `draw`, ktorej obsahom sú funkcie prostredia `canvas`. Aktuálne la-



Obr. 4.6: Ukážka zobrazenia ladičky v prípade nenaladenej struny (vľavo) a v prípade úspešného naladenia struny (vpravo)

dený tón je zobrazený najväčším písmom v strede dolnej časti ladičky (pre predstavu aké ostatné struny sú potrebné k ladeniu, sú ich názvy tónov vyjadrené z jednotlivých strán). Nad týmto nápisom je možné vidieť aktuálnu výšku zahraného tónu v Hertzoch. Najdôležitejšia časť ladičky je ale hlavný ukazovateľ červenej farby, pomocou ktorého je možné detekovať odchýlku od skutočnej frekvencie, ktorá náleží ladenému tónu. Rovnako ako pri porovnávaní hľadanej frekvencie s tou zahranou pri tabulatúre je potrebné aj v tomto prípade počítat s istou odchýlkou. Táto odchýlka je vo veľkosti maximálne 1 centu, čo odpovedá približne 0,1 Hz[23].

## Kapitola 5

# Užívateľská prívetivosť v aplikácii

User experience (UX) design je súbor metód a zásad určených k návrhu produktov zameraných na kvalitný užívateľský zážitok. Zaoberá sa návrhom interakcie medzi používateľom a produktom, či už ide o webovú stránku, mobilnú aplikáciu alebo iný digitálny produkt. Okrem spomínaného zlepšenia celkového zážitku používateľa má za úlohu uľahčiť navigáciu v digitálnom prostredí[15].

Pri tvorbe UX dizajnu sa zvyčajne používa množstvo metód a nástrojov, vrátane užívateľských testovaní, výskumu trhu, analýzy dát, wireframingu a prototypovania. Hlavným záujmom je funkčnosť a intuitívne ovládanie užívateľského rozhrania pre potreby spokojného zákazníka a zanechania pozitívneho dojmu.

Zásady správneho UX dizajnu:

- Prispôsobenie obsahu cieľovej skupine
- Každé dôležité rozhodnutie a zmeny je potrebné podložiť reálnymi užívateľskými dátami
- pravidelná optimalizácia, aktualizácia softwaru a udržiavanie aktuálnych trendov
- držanie sa jednotného štýlu vrámci celej aplikácie

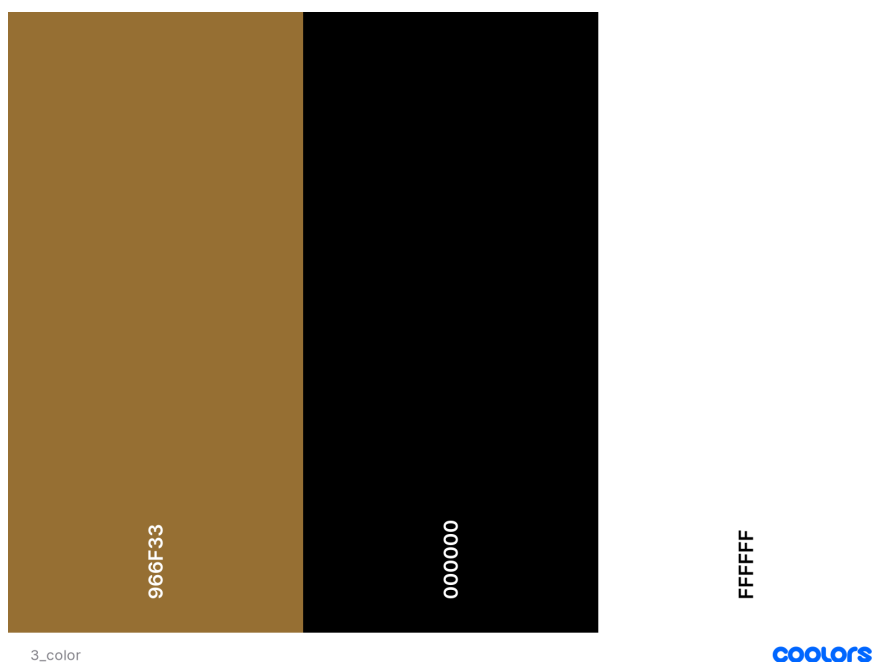
### 5.1 Využitie zásad správneho UX vo webovej aplikácii

Správne použitie zásad UX dizajnu vo webovej aplikácii môže viesť k zlepšeniu užívateľského zážitku a celkovému zlepšeniu užívateľskej angažovanosti s produktom.[18] Niektoré zásady UX dizajnu, ktoré môžu byť uplatnené vo webových aplikáciách sú:

- **Jednoduchosť** - Aplikácia by mala byť pre používateľov jasná a jednoduchá na používanie. To znamená, že by mala byť ľahko navigovateľná a používateľsky prístupná. Navigačné prvky, ako sú menu a tlačidlá, by mali byť jasne viditeľné a ľahko ovládateľné.
- **Konzistencia** - Konzistentný dizajn a užívateľské rozhranie patria k nepostrádateľným pravidlám užívateľskej prívetivosti pri návrhu a implementácii webových aplikácií. To znamená, že prvky by mali byť umiestnené vždy na rovnakom mieste a používať sa rovnaké farby, fonty a vizuálne prvky. Toto pomáha používateľom orientovať sa v aplikácii a zlepšuje celkový užívateľský zážitok.

- **Personalizácia** - Prispôsobiteľnosť potrebám používateľa. To znamená, že by mala poskytovať možnosti prispôsobenia napríklad jazyka, farieb a layoutu podľa daných užívateľských preferencií.

V prvom rade je potrebné vsadiť na prvý dojem. Na docielenie prvotnej pozitívnej reakcie bolo použité pravidlo **3 dominantných farieb** pre našu aplikácie v približnom pomere **6:3:1**. Podľa tohto pravidla by malo byť 60% použitej farby hlavná farba, 30% vedľajšia farba a 10% tzv. akcentová farba[6]. Nasadenie vhodných nástrojov a ich rozmiestnenie v



Obr. 5.1: Prehľad dominantných farieb využitých v tejto webovej aplikácii v pomere 6:3:1 vypracovaných pomocou aplikácie Coolors

našej aplikácií vychádza z nášho užívateľského prieskumu z kapitoly 2.2.

Je potrebné dbať aj na majoritnú užívateľskú vzorku, ktorú tvoria prevažne užívatelia, ktorí majú elementárne hudobné znalosti a aplikácia musí byť jednoducho použiteľná a nemala by používať zbytočne náročnú hudobnú terminológiu či príliš veľké množstvo funkcií, ktoré by mohli byť pre používateľa patriaceho do takejto skupiny mätúce.

Vrámci ďalších sekcií tejto kapitoly sa pozrieme na prevedenie UX návrhov pre desktopovú a mobilnú verziu pomocou tzv. **wireframov**.

Wireframe predstavuje najzákladnejšiu kostru webovej stránky, naznačuje jej štruktúru, rozloženie prvkov, funkčnosť a charakterizuje jednotlivé kontajnery (časti stránky). Na wireframe sa môžete pozerat ako na akýsi náčrt toho, ako bude webová stránka vyzerat. Na jeho základe je možné pochopiť základné rozloženie stránky a jeho užívateľských funkcionalít.[4] Na rozdiel od návrhu webu sa wireframe nezaobera konkrétnou podobou prvkov, ako sú farby, tvary alebo obrázky a ilustrácie.

Wireframe je prvý krok pri návrhu webovej stránky / aplikácie.

Ďalší aspekt, ktorý je potrebný dodržať vrámci našej aplikácie je praktická interakcia s užívateľom počas gitarovej hry. Keďže samotné využitie počítačovej myši je v kombinácií s hraním na gitare nežiadúce, bolo potrebné využiť možnosti, ktoré ponúkajú implementácie automatického posunu zobrazenej časti tabulatúry na obrazovke v reálnom čase.

V prípade, že sa posuvný kurzor zobrazujúci aktuálne miesto na tabulatúre na ktoré by užívateľ mal reagovať dostane do spodnej časti obrazovky, je pomocou animačných prostriedkov a možností daného webového prehliadača automaticky zobrazená časť tabulatúry, ktorá predtým nebola na obrazovke vidieť.

Táto funkcionality bola implementovaná vo Vanilla JS spôsobom, popísaným pod kódom zo zdrojového súboru `tuner.js`.

```
const currentElement = document.getElementById("moveableArrow");

// Get the position of the current element relative to the viewport
const elementRect = currentElement.getBoundingClientRect();

// Scroll the page to the position of the current element
window.scrollTo({
  top: elementRect.top + window.pageYOffset - 100,
  left: elementRect.left + window.pageXOffset,
  behavior: "smooth", // Add smooth scrolling effect
});
```

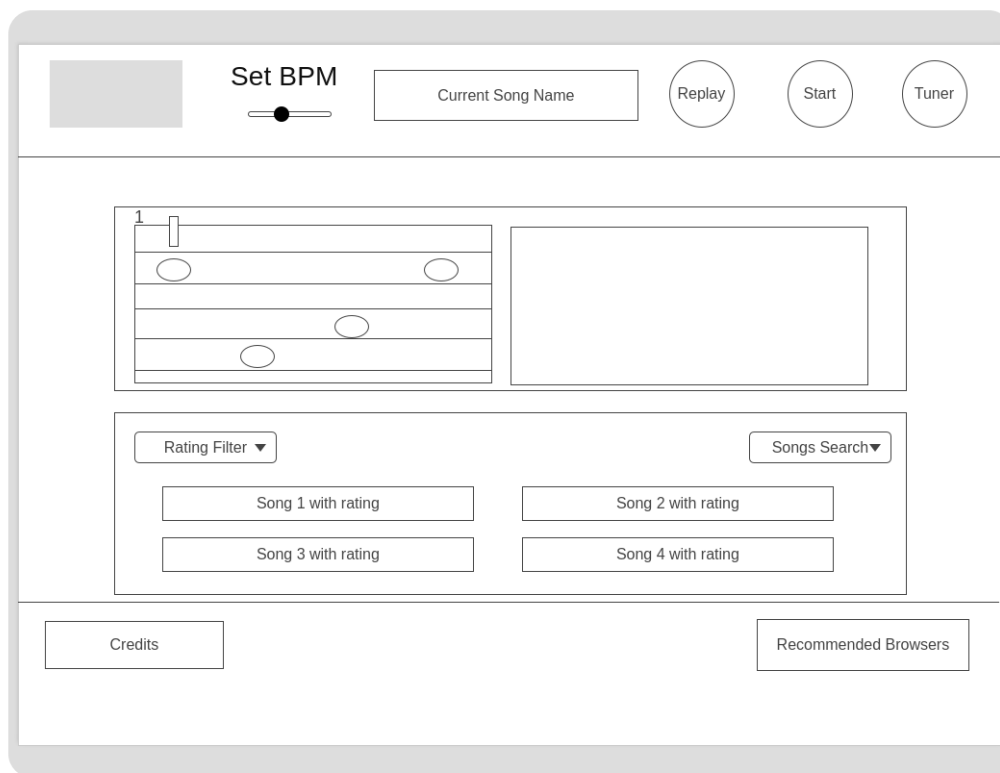
`currentElement` je konštanta do ktorej je vložený objekt, popisujúci element vyjadrujúci posuvný ukazovateľ vrámci tabulatúry. Následne do konštanty `elementRect` uložíme aktuálnu pozíciu tohto elementu a následne pomocou metódy `scrollTo` bude aplikácia posunutá na aktuálnu pozíciu tohto kurzoru (vertikálny posun je znížený o 100 pixelov aby boli viditeľné ostatné časti tabulatúry, ako napr. číslo taktu).

## 5.2 Uživateľská prívetivosť pre desktopovú verziu

Pri návrhu UX pre desktopovú verziu webovej aplikácie je potrebné zohľadniť niektoré zásady UX designu, ktoré zlepšia celkový užívateľský zážitok. Najdôležitejšie z týchto zásad implementovaných aj v našej aplikácii sú:

- **Umiestnenie prvku** - Prvky ako menu, tlačidlá, obrázky atď. by mali byť umiestnené tak, aby boli ľahko viditeľné a ovládateľné pre používateľov. Napríklad menu by malo byť umiestnené na vrchu alebo na boku obrazovky a tlačidlá by mali byť dostatočne veľké, aby boli ľahko ovládateľné.
- **Responzívny dizajn** - Webová aplikácia by mala byť navrhnutá s ohľadom na rôzne veľkosti obrazoviek. To znamená, že by mala byť responzívna a prispôbena pre ich rôzne veľkosti, aby bolo možné používať aplikáciu na rôznych typoch zariadení.
- **Jednoduchosť a prehľadnosť** - Aplikácia by mala byť pre používateľov jednoduchá a prehľadná. To znamená, že by mala byť ľahko navigovateľná a používateľsky prístupná. Pri navrhovaní používateľského rozhrania by sa malo zohľadniť aj množstvo informácií, ktoré sa zobrazujú, aby neboli používatelia zahltení nepotrebnými informáciami, ktoré môžu mať negatívny dopad na celkový užívateľský zážitok z danej aplikácie.

Tieto UX zásady boli využité pre vytvorenie wireframu desktopovej verzie tejto aplikácie. Keďže hlavnú funkcionálnosť aplikácie (funkcie spojené s tabulatúrou a ladičkou) je potrebné



Obr. 5.2: Návrh desktopovej verzie aplikácie pomocou wireframu, kde je možné vidieť rozloženie webovej aplikácie a jej funkcionalít

mať stále k dispozícii počas celej doby využívania aplikácie, sú tieto funkcie prístupné cez tlačidlá v navigačnej lište aplikácie. Navigačná lišta (Navbar) má nastavenú pozíciu vrámci aplikácie na `position: sticky` a `top: 0`. Kvôli týmto css nastaveniam je navbar stále užívateľovi zobrazovaný s celou paletou funkcií v hornej časti aplikácie, a to aj pri manuálnom alebo automatickom scrollovaní.

Hlavný obsah návrhu je rozdelený do 2 sekcií, a síce na tabulatúru (tá je vygenerovaná po kliknutí na danú skladbu) a prehľadovú sekciu s jednotlivými skladbami a možnosťami jej filtrácie podľa obtiažnosti, prípadne možnosť priameho dohľadania skladby.

Počet zobrazených taktov vrámci tabulatúry, ako aj počet zobrazených tlačidiel s názvami skladieb v jednom riadku je daný veľkosťou obrazovky (napr. počet taktov zobrazených v jednom riadku je od 1400px zvýšený z pôvodných 2 na 3 zobrazené takty).

### 5.3 Uživatelská prívetivosť pre mobilné zariadenia

Mobilné UX sa týka subjektívnej skúsenosti, ktorú zdieľa užívateľ s mobilnou aplikáciou. Tá zahŕňa pozitívne aj negatívne emócie.

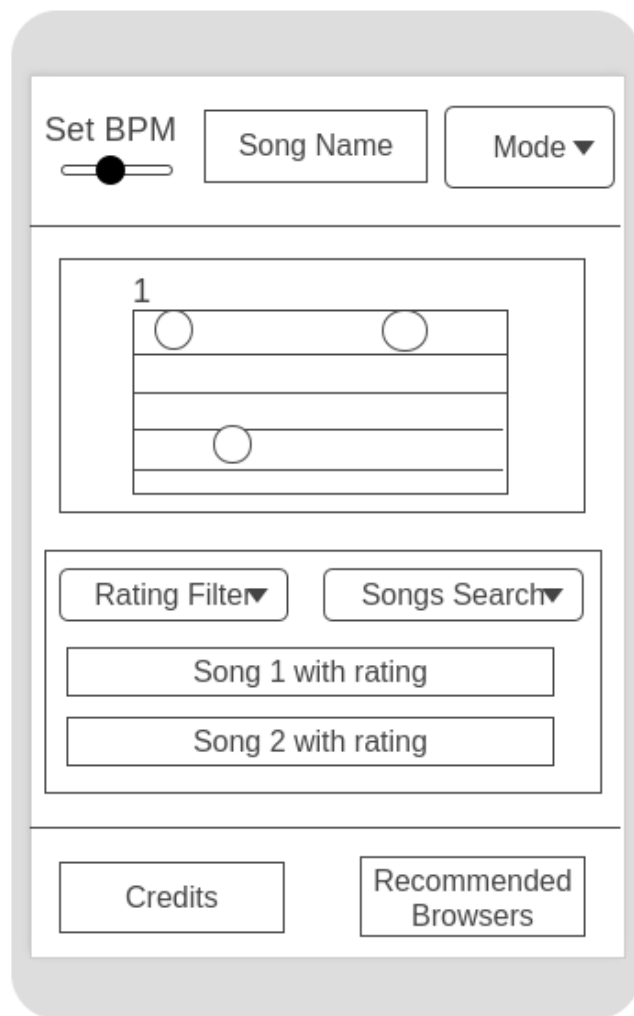
Dobre navrhnutý UX má za cieľ finálny produkt, ktorý je príjemný na používanie. V návrhu mobilného UX je dôležité zvážiť celú škálu zákazníckych potrieb, vrátane priamej interakcie s jej obsahom.[7]

Je potrebné si uvedomiť rozdiel medzi UX a UI, pretože UX návrh zahŕňa oveľa viac ako len grafické prvky. Návrh mobilnej užívateľskej aplikácie sa venuje emóciám, cieľom a výzvam používateľov, a preto presahuje grafické zobrazenie.

Pri jej návrhu, narozdiel od návrhu desktopovej verzie je potrebné dbať na tieto aspekty:

- **Zobrazovacia veľkosť** - Zobrazovanie iba tých najdôležitejších dát, ktoré sa musia v mobilnej verzii zobraziť. Ak tam niečo nemusí byť, takúto informáciu buď vynecháme úplne, alebo im priradíme iné miesto pri zobrazení( pod menu alebo na inú podstránku).
- **Spôsob zobrazenia** - Narozdiel od desktopových verzií, je potrebné dbať aj na vertikálne zobrazenie pre mobilné zariadenia. Horizontálny režim je zvyčajne vhodnejšie pre úlohy, ktoré zahŕňajú čítanie a posúvanie obsahu, zatiaľ čo vertikálny je vhodnejší pre aplikácie, ktoré zahŕňajú sledovanie videí alebo hranie hier.

Najväčšie zmeny oproti desktopovej verzii sú v obmedzenom priamom zobrazení jednotlivých hlavných funkcií. Podľa priloženého wireframu v tejto sekcii je možné v pravom hornom rohu vrámci navbaru vidieť návrh formou `selectboxu`. Tento element obsahuje možnosti výberu jednotlivých funkcionalít, ktoré sú obdobné ako pri prístupe cez desktopovú verziu. Zároveň vzhľadom na obmedzenú veľkosť mobilných zariadení s cieľom zanechať dáta dostatočne viditeľné je vrámci tabulatúry zobrazený práve jeden takt pre jeden riadok a obdobne to platí aj pri zobrazení jednotlivých skladieb s jej priloženými obťažnosťami.



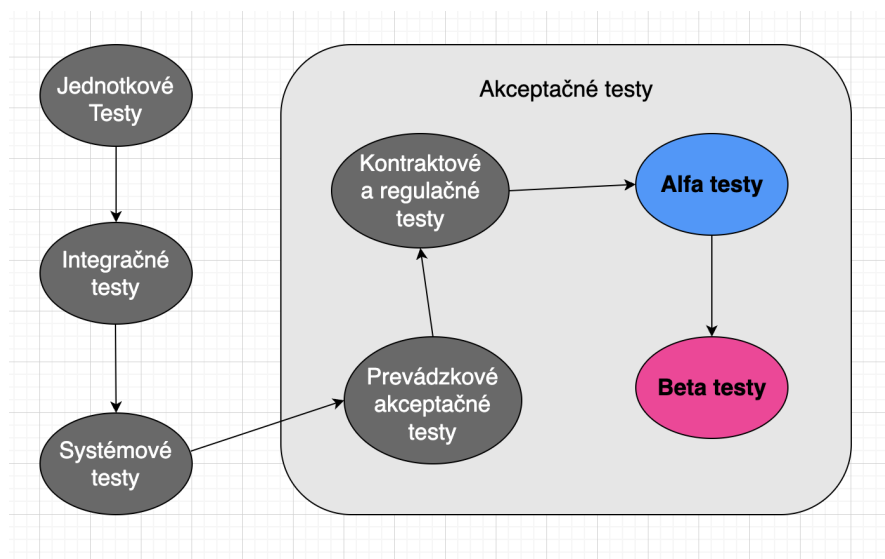
Obr. 5.3: Návrh mobilnej verzie aplikácie pomocou wireframu, kde je možné vidieť rozloženie webovej aplikácie a jej funkcionalít

## Kapitola 6

# Testovanie aplikácie

Aplikácia je primárne určená pre širokú verejnosť bez predošlých predispozičných skúseností. Takéto užívateľské spektrum častokrát nemá skúsenosti s inými obdobnými aplikáciami, tj. je potrebné aby užívateľské prostredie bolo pripravené na neočakávané situácie a stavy, do ktorých sa takýto užívateľ môže dostať.

Daní testerí budú aplikáciu analyzovať podľa predpripravených inštrukcií. Podľa nich bude možné detekovať nežiadúce chyby a následne ich v maximálnej podobe eliminovať. Na základe výstupných dát vytvorených testovaním aplikácie pomocou jednotlivých bodov predpripravenej inštrukčnej sady bude vyhotovený konečný výstupný záznam, podľa ktorého bude možné predikovať aktuálny stav aplikácie a jej pripravenosť na jej následné produkčné nasadenie.



Obr. 6.1: Prehľad jednotlivých krokov pri testovaní webovej aplikácie

Scenáre k výkonu testov pripravuje zákazník spoločne s dodávateľom. Nájdené nezrovnalosti medzi aplikáciou a špecifikáciou sú vrátené vývojáorskemu tímu, a po oprave sú nasadené v zákazníckom prostredí.

Zákazník väčšinou chyby očakáva a preto je vhodné dohodnúť postup, ktorými budú chyby reportované vývojáorskemu tímu a následne v čo najkratšej dobe opravené.

Vzhľadom na veľkosť tejto aplikácie bude tradičný testovací cyklus skrátený a budú využité iba Alfa a Beta testy, ktorých metodický postup a výsledky sa nachádzajú v nasledujúcich sekciách tejto kapitoly.

## 6.1 Alfa testovanie

Alfa testovanie je typ testovania softvéru, ktorý sa vykonáva skôr ako beta testovanie a zameriava sa na odhalenie chýb a nedostatkov vo vývojovej fáze. Tento typ testovania sa vykonáva v skupine testovacích subjektov, ktorí sú zvyčajne interní zamestnanci alebo zákazníci vývojárskej spoločnosti.

Táto testovacia metóda webovej aplikácie zahŕňa testovanie funkcionalít aplikácie, použiteľnosti, bezpečnosti a výkonu. Testovanie sa zvyčajne vykonáva v uzavretej skupine a pod dohľadom vývojárskeho tímu, aby sa zabezpečilo, že sa získajú užitočné informácie pre vylepšenie aplikácie. Medzi testovacie scenáre môžu patriť rôzne akcie a procesy, ktoré používatelia zvyčajne vykonávajú vrámci webovej aplikácie ako aj subjektívne zhodnotenie užívateľského dojmu z danej aplikácie.

Po ukončení alfa testovania a vyriešení chýb, ktoré sa vyskytli, sa môže webová aplikácia posunúť do beta testovania, kde sa testovanie rozšíri na širšie spektrum používateľov, aby sa overilo, či aplikácia funguje dobre v reálnych podmienkach a pre rôzne typy používateľov. Alfa testovanie vrámci tejto aplikácie bolo vykonané vrámci študentskej konferencie vedy a techniky v oblasti informačných technológií v priestor fakulty informačných technológií Vysokého Učení Technického s názvom Excel FIT.

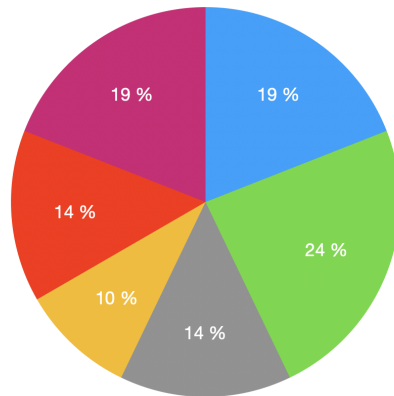
Vrámci tejto konferencie malo niekoľko desiatok záujemcov možnosť vyskúšať si webovú aplikáciu v praxi, zhodnotiť jej aktuálny stav a navrhnúť úpravy aktuálne implementovaných nástrojov. Aplikácia bola predstavená vrámci zariadenia a v prostredí v ktorom bola vyvíjaná, čiže všetky nasledujúce výsledné hodnoty boli zdokumentované pri použití v známom prostredí.

Po predstavení práce a jej následnej praktickej ukážke sa väčšina respondentov zhodla na nasledujúcich pozitívnych bodoch a výhodách aplikácie:

- Užívateľské prostredie pôsobí dobrým dojmom a pre začiatočníkov a mierne pokročilých sa v nej užívateľ vie bez výhrad dobre orientovať
- Možnosť úpravy tempa danej skladby pre optimálny nácvik podľa potrieb užívateľa
- Výborná odozva na okolitý hluk a šum vďaka robustnosti algoritmu CREPE (priemerný nameraný okolitý hluk aplikáciou Youlean bol počas využívania aplikácie na detekciu zahraniého tónu z gitary približne 40 dB)

Respondenti taktiež poukázali na určité nedostatky aplikácie, ktoré nadobudli pri jej testovaní používaní alebo pri prezentácii práce a jej následnej praktickej ukážke. Jednotlivé body sú predstavené na koláčovom grafe vrámci tejto sekcie. Tieto skutočnosti boli pri ďalšom vývoji a implementácii aplikácie zohľadnené a komentár k týmto pripomienkam sa nachádza pod koláčovým grafom.

- Implementácia akordov
- Možnosť zastavenia aplikácie pri zle zahranom tóne a počkanie na jeho správne zahrание
- Zobrazovanie ladičky aj pri prvotnom vybratí skladby
- Zmena designu a veľkosti ukazovateľa aktuálnej pozície v skladbe
- Ukazovateľ skladby by mal mať posun vrámci jednej doby a nie vrámci daných tónov skladby
- Vhodnejšie znázornenie dĺžky tónov



Obr. 6.2: Prehľad rozdelenia jednotlivých nedostatkov a návrhov k ďalšiemu vývoju aplikácie z celkového množstva 20 respondentov vyjadrených v % podľa zhodných vyjadrení

Podľa koláčového grafu jednotlivých nedostatkov a návrhov od respondentov vyplýva, že najčastejšie sa respondenti zhodli na implementácii možnosti zastavenia aplikácie pri zle zahranom tóne. Tento návrh bol pri ďalšom vývoji webovej aplikácie implementovaný ako súčasť tabulatúrových funkcií. V konečnej verzii aplikácie má pred začiatkom hry užívateľ na výber 2 varianty gitarovej hry:

- Aplikácia detekuje a graficky znázorňuje, ktoré tóny boli odohrané správne a nesprávne bez zastavenia chodu aplikácie, dokým užívateľ fyzicky neklikne na tlačidlo **Pauza**
- Pri detekcii zahrания nesprávneho tónu užívateľom sa aplikácia pozastaví a čaká, dokým užívateľ nezahrá tón v očakávanej frekvenčnej výške

Ďalšia z chýbajúcich funkcionalít je detekcia akordov vrámci skladieb v aplikácii (aktuálna verzia aplikácie je schopná detekovať iba jeden tón v daný okamih).

Vzhľadom na pomerne komplexnú problematiku bol tento bod a jeho návrh implementácie popísaný v 8 kapitole tejto práce.

Jedným z často spomínaných nedostatkov bol nedostatočne výrazný, posuvný ukazovateľ aktuálneho miesta v hranej skladbe. Pôvodne bol vyzobrazený znakom ↓, ktorý bol v ďalšej fáze vývoja aplikácie vývoja nahradený výraznejším prvkom, ktorý narozdiel od predošlého znaku priamo zasahuje svojou veľkosťou do tabulatúry, čím zvyšuje praktický význam vrámci daného nástroja. Zároveň bol zmenený časový interval v ktorom sa tento ukazovateľ zobrazoval na jednotlivých miestach nad tabulatúrou počas doby prehrávania a hry danej skladby. Časový interval bol zmenený z pôvodného zobrazovania sa nad každým tónom skladby na každú jednu dobu.

Posledná z pridaných funkcionalít na základe výsledkov testovania počas konferencie bola automatické zobrazenie ladičky pri prvotnom vybratí skladby užívateľom (pôvodne bolo zobrazenie ladičky iba pomocou kliknutia na tlačidlo v pravom hornom rohu aplikácie).

## 6.2 Beta testovanie

Beta testy sú považované za najefektívnejšie a je pri nich odhalených veľké množstvo skrytých chýb. Príliš veľké zdržanie v tejto úrovni testovania môže mať fatálne následky na úspech celého projektu. Z tohto dôvodu je vhodné si dopredu určiť kritéria, ktoré by mali pokryť funkcionality celej aplikácie a zároveň sa zamerať na jej jednotlivé časti, pri ktorých by mohlo dochádzať k skrytým chybám. Najdôležitejšie kroky pri Beta testovaní:

- **Plánovanie** - Je nutné riadne plánovanie, ktoré definuje ciele testovania so všetkými podrobnosťami, ako je riadenie testu, počet účastníkov daného testovania, čas na testovanie aplikácie, určenie deadlinov a pravidelných meetingov s testujúcimi účastníkmi a ďalšie dôležité podrobnosti týkajúce sa aplikácie
- **Nábor jednotlivých účastníkov testovania** - Pri výbere účastníkov je dôležité klásť dôraz na výber rozmanitého druhu testovacích subjektov. Iba tak je možné zabezpečiť dostatočnú variabilitu výsledkov a tak zvýšiť prínos testovania pre webovú aplikáciu ako takú.
- **Vydanie softwarového manuálu** - Testovací manuál iba mal byť rozdelený do častí, v ktorej jedna sa zameriava na konkrétne úkony od ktorých sa očakáva na narazenie na rôzne problémy podľa typu testujúceho subjektu. Následne je dôležité zanechať pri testovaní dostatočnú mieru voľnosti, keďže pri takejto činnosti dochádza najčastejšie k nájdeniu kritických chýb, ktoré by testovaním priamymi postupmi nemuseli byť nájdené.
- **Zbieranie výsledkov a analýza spätnej väzby** - V tejto časti je potrebné klásť dôraz na popisovanie detailov v prípade, že aplikácia v istých bodoch nespĺňa podmienky, ktoré garantuje. Cieľom je upozorniť developerský team na možné problémy, prípadne poukázať na udalosti, ktoré nemuseli byť vôbec predmetom diskusie pri prvotnej implementácii danej webovej aplikácie

Beta testovanie tejto aplikácie prebiehalo nasledujúcim spôsobom:

Bolo vybraných sedem potencionálnych užívateľov, ako z radov programátorov, tak z radov bežných užívateľov nezainteresovaných v oblasti informačných technológií, ktorí mali k dispozícii akustickú gitaru pre účely praktického testovania aplikácie. Následne im bola zaslaná posledná stabilná verzia webovej aplikácie, ktorú mali testovať na základe priloženého manuálu s jednotlivými testovacími bodmi. Na toto testovanie bolo zároveň doporučené využiť webové prehliadače, ktoré disponujú pokročilejšími technológiami, tj. nebude dochádzať k neúplnej funkcionalite jednotlivých nástrojov rámci webovej aplikácie (v našom prípade išlo primárne o prehliadače Firefox a Safari).

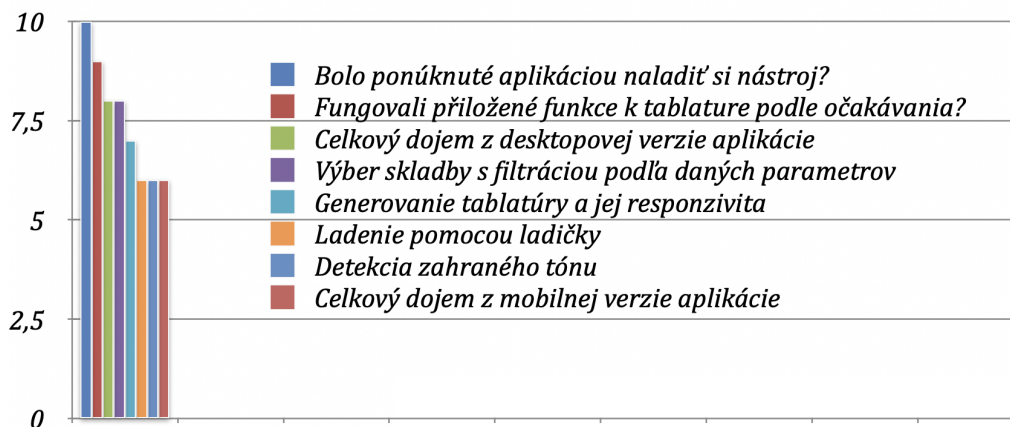
Manuál obsahoval nasledujúce body:

1. Zoznámenie sa s aplikáciou na rôznych zariadeniach (desktopová a mobilná verzia webovej aplikácie)
2. Vybranie ľubovoľnej skladby ponúkaných aplikáciou podľa náročnosti a typu gitarového ladenia
3. Bolo pri prvotnom vybratí skladby automaticky ponúknuté aplikáciou naladenie gitary v správnom ladení podľa danej skladby?

4. Boli ste schopní pomocou priloženého ladiaceho nástroja naladiť gitaru?
5. Bola korektne vygenerovaná tabulatúra danej skladby a bolo jej zobrazenie rezponzívne zobrazené vzhľadom na testovacie zariadenie?
6. experimentujte s jednotlivými funkciami tabulatúry. Funguje prehrávanie skladby korektne bez zasekávania alebo iných nežiadúcich obmedzení?
7. Boli pri hraní na nástroj detekované vami zahrané tóny? Bolo možné kedykoľvek počas hrania pozastaviť skladbu prípadne tlačidlom reset ju vrátiť do začiatočného stavu?
8. aký máte celkový pocit z aplikácie? Myslíte si, že je možné sa vďaka týmto nástrojom a im vstavaným funkciám posunúť pri výučbe hry na gitaru?

Na základe týchto bodov bolo možné otestovať funkcionality jednotlivých komponentov aplikácie a zároveň sa zamerať na testovanie formou subjektívneho hodnotenia aplikácie vybranými testovacími subjektmi. Subjektívne hodnotenie celkového dojmu z aplikácie po užívateľskej stránke je vzhľadom na cieľovú skupinu tejto aplikácie nevyhnutné.

Výsledné záznamy od jednotlivých testovacích subjektov sú zhrnuté v nasledujúcom grafe podľa jednotlivých bodov z testovacieho manuálu. Aby bolo možné toto testovanie zhrnúť do celistvého záveru tak výsledky v slovnej podobe boli preložené do hodnotiacich bodov (1 - nespokojivé, 10 - výborné). Rozšírený slovný popis je k nahliadnutiu v Prílohe B tejto práce. Na základe týchto výsledkov môžeme konštatovať, že aplikácia je stabilná a funguje s



Obr. 6.3: Zjednodušené výsledky Beta testingu na vzorke 7 subjektov

istými výhradami podľa očakávaní. Najväčšie nedostatky boli zaznamenané pri zobrazovaní aplikácie v mobilnej verzii a pri detekcii zahraných tónov z hudobného nástroja. Problémy s detekciou tónu boli spôsobené v nastavení intervalu, v rámci ktorého aplikácia detekovala zahraný tón a následne ho vyhodnocovala. Z toho vyplýva, že je potrebné tento interval rozšíriť aby bolo umožnené užívateľovi triafať jednotlivé tóny s väčšou toleranciou. Zároveň je ale nutné nájsť istú hranicu tejto tolerancie, aby nezasahovala do rozsahu okolitých tónov.

## Kapitola 7

# Vývoj aplikácie do budúcnosti

Ďalší vývoj aplikácie bude závisieť na mnohých faktoroch. Technologický pokrok v oblasti rozpoznávania a detekcie zvuku bude určite jeden z týchto faktorov. Samotná aplikácia aktuálne využíva pár rokov starý algoritmus rozpoznávania dominantnej frekvencie, ktorý má ešte stále priestor na zlepšenie, ako po stránke tréningu na väčšom datasete, tak z hľadiska rýchlosti samotného algoritmu v rozpoznávaní dominantnej frekvencie.

### 7.1 Návrh implementácie detekcie akordov vrámci real-time tabulatúry

Najväčší nedostatok aktuálne vyvinutej aplikácie súvisí s obmedzenými možnosťami algoritmu CREPE. Keďže jeho hlavnou prioritou je detekcia dominantnej frekvencie daného vstupného zvukového signálu nie je možné odchytať sekundárne alebo terciárne frekvencie, ktoré sú pri detekcii zahraniého akordu nevyhnutné. Samotný akord je označenie pre súzvuk troch a viac súčasne znejúcich rôznych tónov. Takto zložený tón je kombináciou mnohých jednoduchých sínusových zložiek, z ktorých každá má vlastnú frekvenciu, amplitúdu a fázu kmitania[22].

Jedným z aktuálne dostupných riešení je detekcia akordov na základe chromagramu. Vytvárajú ho jednotlivé chromatické funkcie (nazývané aj ako profily tónových tried). Takéto funkcie je možné získať zo spektrogramu tým, že sa lineárna frekvenčná os (meraná v Hz) prevedie na logaritmickú os (meranú v tónoch). Výsledná reprezentácia sa nazýva logaritmický spektrogram[20].

Nech  $x$  je diskretný signál s hodnotami patriacimi do množiny reálnych čísel, získaný rovnomerným vzorkovaním s vzorkovacou frekvenciou  $F_s$  udanou v Hz. Ďalej nech  $X$  je Krátkodobá Fourierova transformácia (STFT) s oknom  $w$  dĺžky  $N \in \mathbb{N}$  a krokom  $H \in \mathbb{N}$ . Použitím nulového doplnenia môžeme predpokladať, že Fourierove koeficienty  $X(n, k)$  sú indexované parametrami rámca  $n \in \mathbb{Z}$  a frekvenčnými parametrami  $k \in [0 : K]$ , kde  $K = \frac{N}{2}$  je frekvenčný index zodpovedajúci Nyquistovej frekvencii.

$$F_{coef}(k) = \frac{k \cdot F_s}{N}$$

Hlavnou myšlienkou logaritmického spektrogramu je definovať frekvenčnú os tak, aby zodpovedala logaritmicky rozloženej frekvenčnej distribúcii rovnomerne temperovanej stupnice[8]. Identifikácia tónov sa vykonáva pomocou MIDI číselných zápisov tónov (kde napr. tón A4 zodpovedá MIDI číslu  $p = 69$ , a stredové frekvencie sú určené nasledujúcou rovnicou:

$$F_{pitch}(p) = 2^{(p-69)/12} \cdot 440$$

Na základe vyššie spomínanej metódy by bola možná implementácia efektívneho nástroja na detekciu viacúrovňových dominantných frekvencií s možnosťou prípadného zvýraznenia jednotlivých tónov, ktoré by pri zahraní na gitare neboli dostatočne znelé (dominantné), čím by bol hráč priamo upozornený na chybné zahratie akordu vrámci danej piesne.

## 7.2 Spôsobý pridávania nových piesní do aplikácie

Do aktuálnej verzie aplikácie je možné pridávať jednotlivé hudobné taby kombináciou ich umiestnenia na tabulatúre - gitarová struna a pozícia tónu na gitarovom pražci (napr. H4) a jej dĺžkou (pre štvrtou notu je to 1).

Keďže bude v budúcnosti potrebná čoraz väčšia databáza hrateľných piesní, vo forme tabulatúry sú aktuálne k dispozícii tieto 2 formy riešenia popísané v podsekciiach tejto sekcie.

### 7.2.1 Integrácia pomocou API z dostupných webov s databázami tabov

Táto forma je jednoznačne najjednoduchšia možnosť, ako dostať do databázovej sekcie aplikácie najväčšie množstvo dát pre piesne rôznych žánrov a obtiažností.

Jednou s najznámejších a obsahovo najbohatších webových databáz pre tabulatúry piesní je aplikácia Ultimate Guitar. Keďže táto aplikácia je riadená prevažne verejnou komunitou je jedna z možností využiť aktuálne dostupné programovateľné aplikačné rozhranie (API). Odkaz na zdrojový kód a využitie jedného z týchto aplikačných rozhraní sa nachádzal vrámci komunitného fóra aplikácie Ultimate Guitar. Tento príspevok bol pridaný v roku 2019 od používateľa so pseudonymom joncardasis. Toto API slúži primárne na získanie akordov z daného zdroja, preto bude potrebné rozhranie rozšíriť aj o možnosť stiahnutia dát pre tabulatúru.

Dáta je možné stiahnuť pomocou funkcie na ktorú je potrebné sa dopytovať pomocou `tab([URL tabulatúry pod doménou Ultimate Guitar])`. Následne tieto dáta budú po úprave uložené do dátovej štruktúry vhodnej pre našu aplikáciu (spomínané na začiatku tejto sekcie). Keďže väčšina tabov z týchto webov (najmä ak sa jedná o známejšie piesne) obsahovo sedia k piesni, ktorú reprezentujú, sú takto získané dáta vhodnou možnosťou, ako niekoľkonásobne zvýšiť celkový počet piesní určených pre koncového užívateľa na gitarovú hru.

## 7.2.2 Rozšírené užívateľské rozhranie pre tvorbu vlastnej tabulatúry

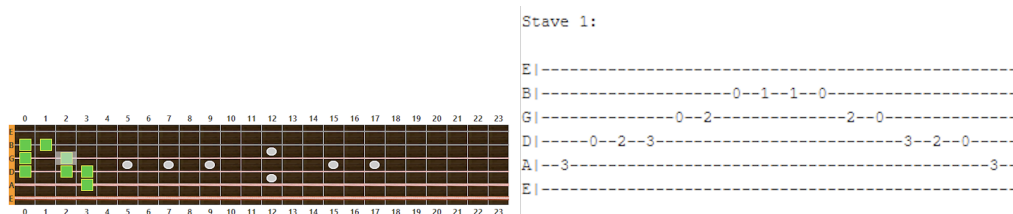
Vhodný variant pridávania nových dát do aplikácie je vytvorenie vlastného rozhrania na pridávanie a úpravu tabulatúry. Pomocou tohto nástroja bude možné upravovať aj ostatné piesne v aplikácii.

Obsahom rozhrania by mali byť nasledujúce tri sekcie:

- Interaktívna a klikateľná verzia gitarového hmatníku
- Výstupná tabulatúra s popisom jednotlivých tónov
- Nevyhnutné, doplňujúce funkcie, ako preskočiť do nového tabu, pridať pomlčku, ...

Takéto nástroje už k dispozícii sú a väčšie hudobno-gitarové aplikácie s podobným zameraním už častokrát práve obdobné rozhrania vytvorené majú kvôli pridávaniu a editácii tabulatúry.

Jedným z voľne dostupných nástrojov je Guitar Tab Creation. Aplikácia bola z užívateľského hľadiska vyskúšaná a jej výsledok je zobrazený nižšie pre lepšie pochopenie zobrazenia jednotlivých funkcií tohto rozhrania.



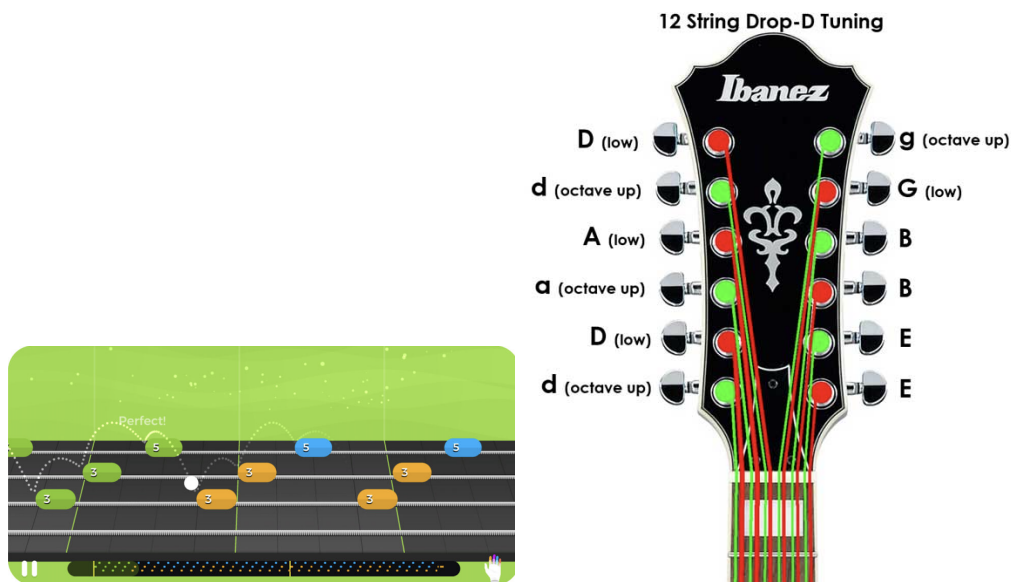
Obr. 7.1: Praktická ukážka využitia aplikácie Guitar Tab Creation

### 7.3 Rozšírenie aplikácie pre rôzne typy akustických gitár

S narastajúcim užívateľským záujmom rastú aj užívateľske požiadavky na aplikáciu, ako takú. Každý užívateľ má rôzne požiadavky a potreby, a preto je vhodné sa s ich narastajúcimi počtami snažiť prispôsobiť takým požiadavkám a potrebám, ktoré sú v záujme čo najväčšieho množstva užívateľov.

Aj z tohto dôvodu je potrebné v budúcnosti zabezpečiť funkcionality webovej aplikácie aj pre iné typy akustických gitár, ako klasickú 6-strunovú akustickú alebo španielsku gitaru ale zamerať vývoj aj smerom k nástrojom ako ukulele, mandolína, či 12-strunová gitara. V prípade, že sa užívateľ rozhodne primárne zamerať svoju hru na spomínané typy akustických gitár je potrebné tejto skutočnosti prispôsobiť aj aplikačné rozhranie:

- zobrazované dáta vrámci tabulatúry
- nastavenie ladičky podľa ladenia daných gitarových strún špeciálne pre typ gitary
- jednotlivé hudobné žánre, ktoré sú najčastejšie hrané s daným typom gitary



Obr. 7.2: Ukážka nastavenia tabulatúry pre ukulele v aplikácii Yousican a variant zobrazenia ladičky pre 12-strunovú gitaru

## Kapitola 8

### Záver

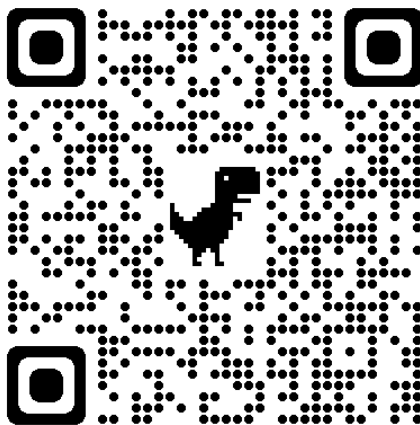
V tejto bakalárskej práci sme najprv zisťovali preferencie našich respondentov na potenciálnu webovú aplikáciu výuky hry na gitaru. Následne sme otestovali niektoré podobné webové aplikácie na trhu, zhodnotili ich výhody a nevýhody pre našu cieľovú skupinu a vytvorili jednoduchú a užívateľsky prívetivú webovú aplikáciu.

Na detekciu dominantnej frekvencie vrámci zvukového signálu bol použitý algoritmus CREPE, kvôli dominancii v jednotlivých experimentoch na konkrétnom datasete oproti konkurenčným algoritmom.

Tento algoritmus bol vrámci JavaScriptovej knižnice ml5.js využitý pri implementácii detekcie výšky zahraného tónu pre nástroje tabulatúra a ladička.

Webová aplikácia najskôr prešla otestovaním na konferencii Excel FIT a následne beta testovaním vďaka 7 testovacím subjektom. Všetky dáta z testovania boli zaznamenané a jednotlivé výsledky boli buď implementované vrámci ďalšieho vývoja webovej aplikácie, alebo bol návrh jej implementácie popísaný v kapitole 7.

Vrámci publikácie bola zakúpená doména `gitaro.tech` spolu so službou Webhosting, kde bola webová aplikácia uložená. Pri používaní aplikácie sa doporučuje pre plnú funkcionality používať prehliadače Firefox, Firefox Developer Edition alebo Safari.



Obr. 8.1: Qr kód odkazujúci na URL webovej aplikácie Gitaro

# Literatúra

- [1] CHE, J. *Construction of Music Intelligent Creation Model Based on Convolutional Neural Network* [[online]]. 2022. Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9276503/>.
- [2] CREPE. *Crepe* [[online]]. 2018. Dostupné z: <https://www.semanticscholar.org/reader/86aeec4d48d949190b3a0c2bf32c101fc23f13a3>.
- [3] DEVELOPER, M. *OscillatorNode* [[online]]. 2023. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/API/OscillatorNode>.
- [4] DUBINSKÁ, L. *CO JE WIREFRAME WEBU, PROČ HO POTŘEBUJETE A JAK HO VYTVOŘIT?* [[online]]. 2022. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-wireframe-predstavujeme-5-duvodu-proc-je-pro-klienty-drateny-model-dulezity>.
- [5] GUITARCOM. *Best software for guitarists in 2022: 12 best apps to learn guitar* [[online]]. 2022. Dostupné z: <https://guitar.com/guides/buyers-guide/best-guitar-learning-apps/>.
- [6] JEPH, P. *An Understanding of Colors for UI Design* [[online]]. 2023. Dostupné z: <https://www.qed42.com/insights/coe/design/understanding-colors-ui-design>.
- [7] KOŘOUSKOVÁ, B. *UX A UI DESIGN: JAK NA UŽIVATELSKÉ ROZHRANÍ WEBŮ A APLIKACÍ?* [[online]]. 2022. Dostupné z: <https://www.rascasone.com/cs/blog/ux-design-ui-design>.
- [8] LABS, A. *Log-Frequency Spectrogram and Chromagram* [[online]]. 2015. Dostupné z: [https://www.audiolabs-erlangen.de/resources/MIR/FMP/C3/C3S1\\_SpecLogFreq-Chromagram.html](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C3/C3S1_SpecLogFreq-Chromagram.html).
- [9] LI, W. *Ml5.js, Machine Learning for Web Pitch detection* [[online]]. 2020. Dostupné z: <https://www.bookstack.cn/read/ml5js/6701022b96707a50.md>.
- [10] LIETAVCOVÁ, B. Z. *CONVOLUTIONAL NEURAL NETWORKS* [[online]]. 2018. Dostupné z: [https://www.vut.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=180844](https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=180844).
- [11] MATTHIAS MAUCH, S. D. *PYIN: A FUNDAMENTAL FREQUENCY ESTIMATOR USING PROBABILISTIC THRESHOLD DISTRIBUTIONS* [[online]]. 2014. Dostupné z: <https://www.eecs.qmul.ac.uk/~simond/pub/2014/MauchDixon-PYIN-ICASSP2014.pdf>.

- [12] MIKUŠOVÁ, A. *Metodika hry na gitaru pre prípravný a 1. ročník na ZUŠ* [[online]]. 2014. Dostupné z: [https://is.jamu.cz/th/ea75a/Bakalarska\\_praca\\_word\\_nova.pdf](https://is.jamu.cz/th/ea75a/Bakalarska_praca_word_nova.pdf).
- [13] ML5. *Js lib* [[online]]. 2022. Dostupné z: <https://ml5js.org/>.
- [14] MUSESCORE. *Tablatura* [[online]]. 2017. Dostupné z: <https://musescore.org/sk/handbook/2/tablatura>.
- [15] NORMAN, D. *The Definition of User Experience (UX)* [[online]]. 2018. Dostupné z: <https://www.nngroup.com/articles/author/don-norman/>.
- [16] PLUS, V. *Asynchronní zpracování v JavaScriptu* [[online]]. 2016. Dostupné z: <http://visionplus.cz/tutorialy/async-javascript.php>.
- [17] SOUND, M. *Monoaural sound* [[online]]. 2023. Dostupné z: <https://sk.theastrologypage.com/monaural-sound>.
- [18] TOUCHIT. *Ako zistím, že je web používateľsky prívetivý?* [[online]]. 2018. Dostupné z: <https://touchit.sk/ako-zistim-ze-je-web-pouzivatelsky-privetivy/173778>.
- [19] UNIZA, F. *REAKČNÝ ČAS* [[online]]. 2020. Dostupné z: [https://fyzika.uniza.sk/wp-content/uploads/2020/08/Reakcny\\_cas\\_objem\\_valceka.pdf](https://fyzika.uniza.sk/wp-content/uploads/2020/08/Reakcny_cas_objem_valceka.pdf).
- [20] VISION, S. C. *Real-Time Music Following in Score Sheet Images via Multi-Resolution Prediction* [[online]]. 2021. Dostupné z: <https://www.frontiersin.org/articles/10.3389/fcomp.2021.718340/full>.
- [21] WIKI hmm. *Tablatúra* [[online]]. 2019. Dostupné z: <https://hmn.wiki/sk/Tablature>.
- [22] WIKIPEDIA. *Akord* [[online]]. 2005. Dostupné z: <https://cs.wikipedia.org/wiki/Akord>.
- [23] ZAJAC, A. *How many cents are in a Hz?* [[online]]. 2017. Dostupné z: <https://harp.andrewzajac.ca/CentsHz>.

## Príloha A

# Obsah priloženého CD

```
/
├── styles/.....Obsahuje zdrojový kód pre kaskádové štýly (CSS)
│   └── style.css.....Doplňujúci zdrojový kód pre časti, ktoré nedoplnil Bootstrap
├── js/ ..... Obsahuje javascriptové súbory s využitím knižníc jQuery a ml5.js
│   ├── tuner.js ..... Javascriptový súbor pre ladičky a detekciu výšky tónov
│   ├── tabs.js ..... Javascriptový súbor pre tabulatúru a detekciu výšky tónov
│   └── general.js ..... Ostatné časti webovej aplikácie (prevažne funkcionality mimo
│       tabulatúry a ladičky)
├── thesis/ ..... Adresár obsahuje technickú správu a jej zdrojové kódy
├── imgs/ ..... Obsahuje obrázky vo formáte PNG použité vrámci aplikácie
├── index.html.....Hlavná HTML stránka vrámci webovej aplikácie
└── README.md
```

## Príloha B

# Rozšírený slovný popis 2 výsledkov Beta testov

**Tereza:** skúšala som aplikáciu s mojou gitarou na svojom notebooku. Najprv som si vybrala skladbu a prehrala ju, to bolo bez problémov (hneď sa mi aj zobrazila ladička podľa ktorej som bola schopná gitaru naladiť). Následne som skúšala verziu s detekciou zahraného tónu z gitary a prišlo mi, že to detekuje o niečo skôr ako by malo. S desktopovou verziou som spokojná, avšak keď som skúšala mobilnú verziu aplikácie prišlo mi, že je hlavne navigačná lišta pomerne rozbitá a trochu sa prekrývajú jednotlivé tlačidlá.

**Marek:** aplikáciu som otestoval na svojom počítači cez doporučený prehliadač Firefox. Keďže už hrám nejaký ten čas na gitare a už som vyskúšal niekoľko podobných aplikácií tak na prvý dojem to vyzerá veľmi jednoducho a príde mi, že sa v tom rýchlo viem zorientovať. Najprv som vyskúšal zmeniť tempo piesne a prehrať si ju aby som zistil v akom tempe to skúsím uhrať (toto išlo bez komplikácií). Následne som vyskúšal funkciu s detekciou tónu a vyšlo mi to celkom dobre (osobne mi prišlo že to detekovalo o niečo skôr) resp. že tá šípka čo je nad tabulatúrou išla moc skoro.