



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název: Mobilní aplikace pro trénink hudebního sluchu
Student: Trong Chung Chau Nguyen
Vedoucí: Mgr. Martin Podloucký
Studijní program: Informatika
Studijní obor: Webové a softwarové inženýrství
Katedra: Katedra softwarového inženýrství
Platnost zadání: Do konce letního semestru 2019/20

Pokyny pro vypracování

Prostudujte problematiku nauky hudebního sluchu a navrhnete efektivní způsob pro trénování sluchu pro uživatele mobilních telefonů a tabletů. Navrhnete a implementujete mobilní aplikaci pro trénink hudebního sluchu, která by umožňovala efektivně trénovat dovednosti jako rozpoznání výšky tónu, určení hudebního intervalu, popř. rozpoznání barvy tónu (hudebních nástrojů) a hudební citění (rozpoznání druhu akordu/melodie/harmonie). Postupujte dle klasického vývojového cyklu software: analýza, návrh, implementace, testování.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 5. února 2019



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalářská práce

Mobilní aplikace pro trénink hudebního sluchu

Trong Chung Chau Nguyen

Katedra softwarového inženýrství

Vedoucí práce: Mgr. Martin Podloucký

16. května 2019

Poděkování

Chtěl bych za vše poděkovat Mgr. Martinu Podlouckému za vedení této práce. Bez něho by tato práce nemohla vzniknout, jelikož byl tím, kdo mě inspiroval ke studiu problematiky tréninku hudebního sluchu prostřednictvím mobilních aplikací. Dále bych chtěl poděkovat svému učiteli ze ZUŠ Klatovy Františku Pelíškovi za odborné konzultace v oblasti hudební nauky a hudebního sluchu. V neposlední řadě bych chtěl poděkovat všem svým blízkým za podporu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 16. května 2019

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2019 Trong Chung Chau Nguyen. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Nguyen, Trong Chung Chau. *Mobilní aplikace pro trénink hudebního sluchu*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Cílem práce je navrhnout hypotetické řešení pro trénování hudebního sluchu. V práci jsou vysvětleny související základní hudební pojmy a jsou porovnány konkurenční mobilní aplikace s navrhovaným řešením.

Následně se práce zabývá klasickým vývojovým cyklem softwaru (analýza, návrh, implementace, testování) pro realizaci mobilní aplikace trénující hudební sluch.

Výsledná mobilní aplikace umožňuje plánování tréninků a spuštění libovolně konfigurovatelných zvuků. Pro implementaci mobilní aplikace byl použitý multiplatformní framework Xamarin.

Klíčová slova hudba, trénink, hudební sluch, trénink hudebního sluchu, mobilní aplikace, Xamarin

Abstract

This thesis proposes a hypothetical solution for musical hearing training and provides clarifications of fundamental music terms. Competing alternatives are compared with the proposed solution.

Furthermore, classic development software cycle (analysis, draft, implementation, testing) are elaborated for the inception of mobile application for musical hearing training.

The final mobile application features training planning and a configurable soundboard. Multiplatform framework Xamarin has been used as a foundation for the application.

Keywords music, training, musical hearing, musical hearing training, mobile application, Xamarin

Obsah

Úvod	1
1 Cíl práce	3
2 Základní hudební teorie	5
2.1 Tón a jeho vlastnosti	5
2.2 Tónová soustava a názvy tónů	6
2.3 Intervaly	8
2.4 Stupnice	8
2.5 Akordy	9
3 Trénování hudebního sluchu	13
3.1 Elementární disciplíny	13
3.2 Strategie pro trénování hudebního sluchu	13
4 Analýza	15
4.1 Funkční a nefunkční požadavky	15
4.2 Vyhodnocení konkurenčních aplikací	16
5 Návrh řešení	23
5.1 Navržené změny oproti konkurenčním aplikacím	23
5.2 Popis funkcionalit aplikace	24
5.3 Návrh uživatelského rozhraní	25
5.4 Architektura	25
5.5 Databázová vrstva	29
5.6 Obchodní logika	31
6 Implementace	33
6.1 Nástroje	33
6.2 Implementace Audio Service	35

7 Testování	39
Závěr	41
Možné rozšíření a vylepšení	41
Seznam použitých zdrojů	43
A Seznam použitých zkratk	47
B Ukázky mobilní aplikace	49
C Obsah příloženého USB disku	53

Seznam obrázků

2.1	Klaviatura a tónová řada	6
2.2	Stupnice C dur	9
2.3	Stupnice A moll	10
4.1	Aplikace The Ear Gym	18
4.2	Aplikace Functional Ear Trainer	20
4.3	Aplikace Perfect Ear	22
5.1	Diagram případů užití	26
5.2	Wireframe aplikace	27
5.3	Architektura MVVM	28
5.4	Databázová vrstva	30
6.1	Visitor pattern – Diagram tříd	36

Seznam tabulek

2.1	Tónové soustavy	7
2.2	Výčet tónů v jednočárkované oktávě	7
2.3	Výčet intervalů	8
2.4	Výčet druhů stupnic	9
2.5	Výčet druhů akordů	11
4.1	Shrnutí konkurenčních řešení	21

Úvod

Tato práce se zabývá problematikou zlepšování hudebního sluchu, která je následně řešena pomocí mobilní aplikace pro jeho trénování.

Hudební sluch je schopnost rozeznat znějící tóny, které slyšíme v hudbě. Podle mého názoru je špatný hudební sluch jako špatný zrak. Představme si, že žijeme se špatným zrakem. Dokonce i ty nejjednodušší úkoly by pro nás byly obtížné. Pak si představme, že jsme si poprvé nasadili brýle. Najednou je vše jasně vidět. Úkoly, které byly pro nás dříve obtížné, jsou nyní jednoduché a snadné. [1]

Když budeme rozvíjet svůj hudební sluch, budeme vnímat hudbu zcela jiným rozměrem, než jsme doposud byli zvyklí. Pomůže nám to v mnoha ohledech zlepšit naše hudební vnímání a sblíží nás to s naším hraním, takže si ho budeme moci užívat více než doposud. [1]

V této práci budou vysvětlené jednotlivé elementy hudby a hudební názvosloví evropské hudby. Poté se práce zaměří na hudební sluch. V práci budou vysvětlené disciplíny trénování hudebního sluchu a na jejich základě bude navržena mobilní aplikace.

Cíl práce

Cílem této práce je analyzovat, navrhnout, implementovat a otestovat funkční prototyp mobilní aplikace pro trénink hudebního sluchu. Nejprve je potřeba čtenáře seznámit se základy hudební teorie, aby mohl porozumět problematice zkvalitňování hudebního sluchu a pochopit využití mobilní aplikace při hudební výuce. Následně je zapotřebí zanalyzovat požadavky na výslednou aplikaci a porovnat existující řešení, ze kterých se částečně odvodí návrh řešení.

Účelem výsledné aplikace je efektivní trénink hudebního sluchu uživatele. Toho bude docíleno nejen díky různým *Learn through play* aktivitám, ale i díky dalším funkcionalitám. Mezi ně patří například generování podrobných statistik s přehledem silných a slabých stránek uživatele, možnost přehrávání libovolných zvuků a v neposlední řadě také přítomnost průvodce, jenž by pomohl uživateli naplánovat a sestavit další lekce individuálně se odvíjející od jeho schopnosti.

Základní hudební teorie

Pro představu o tom, co se trénuje v rámci zkvalitňování hudebního sluchu a jak se hudební sluch dá vytrénovat, je zapotřebí znát základy hudební teorie. V této kapitole bude uvedena pouze nezbytně nutná teorie k pochopení látky, jíž se věnuje výsledná mobilní aplikace. V práci budou vysvětlené definice tónu, jeho základní vlastnosti a systém názvosloví. Dále se bude pokračovat posloupností tónů, které tvoří intervaly nebo stupnice, a nakonec se práce dotkne souzvuku tónů – akordy.

Veškerá hudební teorie v této kapitole včetně tabulek je převzatá z publikace *ABC: hudební nauky* [2] a zjednodušená ve spolupráci s pedagogem Františkem Pelíškem [3].

2.1 Tón a jeho vlastnosti

Zvuk je mechanické vlnění v látkovém prostředí, které je schopné vyvolat sluchový vjem. Zvuky dělíme na tóny a hluky. Tóny vznikají při pravidelném kmitání. Za hluky označujeme nepravidelná vlnění vznikající jako složité nepravidelné kmitání těles nebo krátké nepravidelné rozruchy (srážka těles, výstřel apod.).

V hudbě používáme především tóny, které mají 4 základní vlastnosti:

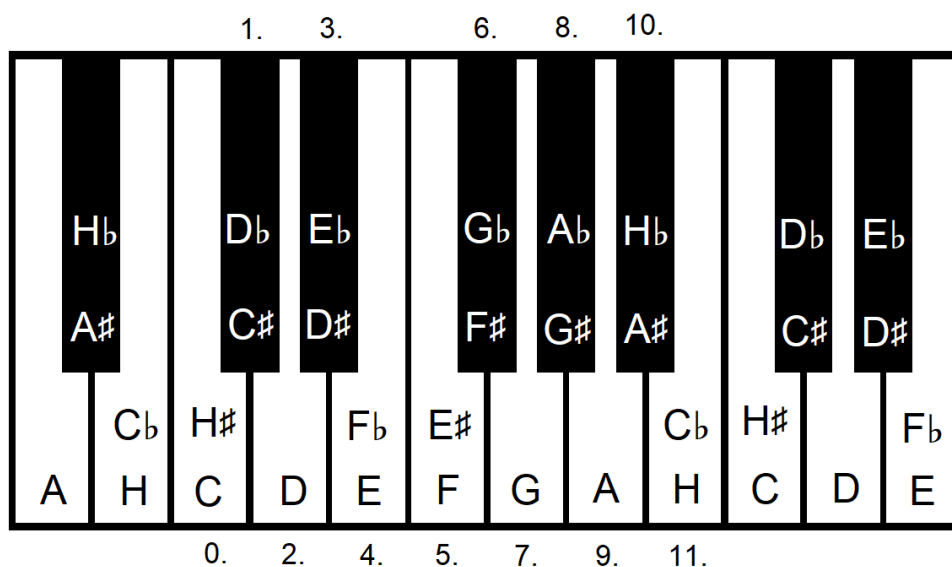
výška udává frekvenci tónu,

síla udává hlasitost tónu¹,

délka udává, jak dlouho tóny znějí,

barva udává charakteristické znění tónu. Tóny se při stejné výšce mohou lišit odlišným zbarvením, a mohou tak vyvolat jiný charakteristický zvukový vjem. Různé barvy tónu označujeme především podle původu tónu, zdrojem může být například hudební nástroj.

¹V hudební nauce se používá síla tónu, v akustice hlasitost.



Obrázek 2.1: Klaviatura a základní tónová řada

2.2 Tónová soustava a názvy tónů

V evropské hudbě se využívá přibližně 120 tónů. Uspořádání všech těchto tónů podle jejich výšek vytváří tónovou soustavu. Jejím základem je 7 tónů: *c*, *d*, *e*, *f*, *g*, *a* a *h*, které se v tónové soustavě několikrát opakují v různých výškových polohách. Souhrnně se nazývají základní tónová řada.

Na obrázku 2.1 jsou vidět tóny rozložené na klaviatuře, která je známa například z klavíru, ale také třeba z akordeonu. Klaviatura se skládá z bílých a černých kláves. V rozmezí jedné oktávy je 12 kláves – 7 bílých a 5 černých. Základní tónová řada je umístěna zleva doprava na bílých klávesách. Černé klávesy se po celé klaviatuře střídají vždy dvě a tři, jak je vidět z obrázku 2.1.

Nejmenší vzdálenost ve výšce dvou tónů se nazývá v naší hudbě půltón a nachází se vždy mezi dvěma sousedními klávesami, a to buď mezi dvěma bílými (např. *e*–*f* nebo *h*–*c*), nebo mezi jednou bílou a sousední černou (např. *f*–*f*[#]). Každý tón naší tónové soustavy můžeme zvýšit nebo snížit o půltón. Dvojitě zvýšení nebo snížení by pak bylo o celý tón. Jednoduché zvýšení se provede přidáním křížku (např. zvýšením tónu *c* se získá tón *c*[#]). Podobně jednoduché je snižování, které se provede pomocí přidání béček (např. snížením tónu *c* se získá tón *c*_b, který je totožný s tónem *h*).

Oktáva se skládá ze 12 půltónů po sobě jdoucích. Pro zjednodušení budeme používat zvýšení tónů a tóny pojmenujeme *c*, *c*[#], *d*, *d*[#], *e*, *f*, *f*[#], *g*, *g*[#], *a*, *a*[#], *h*.

2.2. Tónová soustava a názvy tónů

Název oktávy	Velikost počátečního písmena	Počet čárek v zápisu	Obsažené tóny
Subkontra	velké	2	$C''—H''$
Kontra	velké	1	$C'—H'$
Velká	velké	0	$C—H$
Malá	malé	0	$c—h$
Jednočárkovaná	malé	1	$c'—h'$
Dvoučárkovaná	malé	2	$c''—h''$
Tříčárkovaná	malé	3	$c'''—h'''$
Čtyřčárkovaná	malé	4	$c''''—h''''$
Pětíčárkovaná	malé	5	$c'''''—h'''''$

Tabulka 2.1: Tónové soustavy

Pořadí tónu	Tón	Frekvence
0	c'	261,63
1	c^{\sharp}	277,18
2	d'	293,66
3	d^{\sharp}	311,13
4	e'	329,63
5	f'	349,23
6	f^{\sharp}	369,99
7	g'	392,00
8	g^{\sharp}	415,30
9	a'	440,00
10	a^{\sharp}	466,16
11	h'	493,88

Tabulka 2.2: Výčet tónů v jednočárkované oktávě

Tónová soustava obsahuje celkem devět oktáv, které máme v prvním sloupci tabulky 2.1. Ve druhém sloupci je pravidlo pro značení tónů v dané tónové soustavě. V tabulce 2.2 je výčet tónů v jednočárkované oktávě² s příslušnými frekvencemi v hertzech. Poměr frekvencí mezi tónem a tónem zvětšený o půltón je $1 : \sqrt[12]{2}$.

²Místo čárek se může napsat číslo vyjadřující počet čárek v horním indexu, $c'' = c^2$.

Výšková vzdálenost	Interval	Tóny v základní tónové řadě
0	Čistá prima	$c-c$
1	Malá sekunda	$c-c\sharp$
2	Velká sekunda	$c-d$
3	Malá tercie	$c-d\sharp$
4	Velká tercie	$c-e$
5	Čistá kvarta	$c-f$
6	Zmenšená kvinta	$c-f\sharp$
7	Čistá kvinta	$c-g$
8	Malá sexta	$c-g\sharp$
9	Velká sexta	$c-a$
10	Malá septima	$c-a\sharp$
11	Velká septima	$c-h$
12	Čistá oktáva	$c-c'$

Tabulka 2.3: Výčet intervalů

2.3 Intervaly

Interval je v hudbě výšková vzdálenost mezi dvěma tóny. Základní intervaly (čisté a velké) se odvozují od základní tónové řady. Čisté intervaly jsou *prima*, *kvarta*, *kvinta* a *oktáva*. Velké intervaly jsou *sekunda*, *tercie*, *sexta* a *septima*.

Zmenšením velkých intervalů dostáváme malé. Čisté intervaly až na čistou *kvintu* nezmenšujeme. Zmenšením čisté *kvinty* dostáváme zmenšenou *kvintu* nebo zvětšenou *kvartu*. Výčet intervalů seřazených vzestupně podle výškové vzdálenosti máme v tabulce 2.3.

2.4 Stupnice

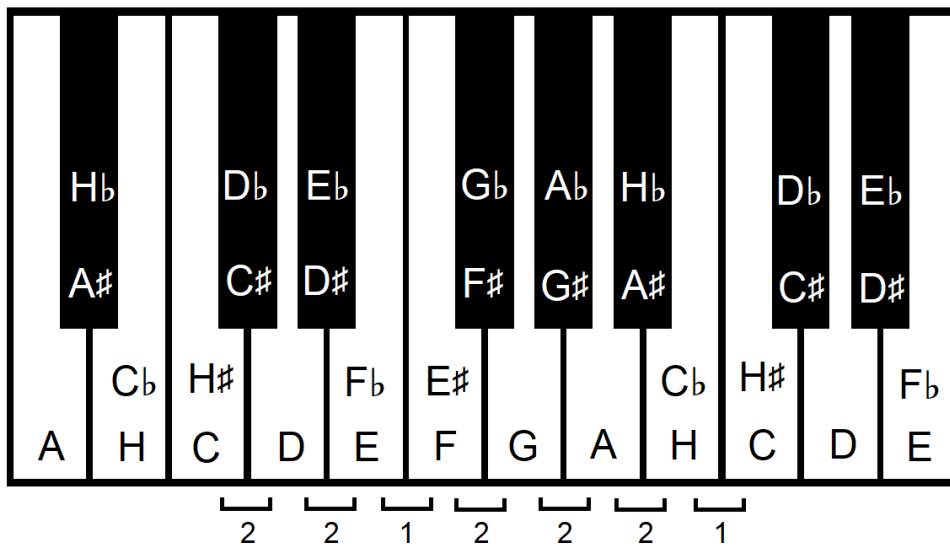
Stupnice je stoupající nebo klesající řada tónů v rozmezí jedné oktávy, uspořádaná podle určitých pravidel. Tato pravidla se týkají především počtu tónů v oktávě a vzdálenosti mezi jednotlivými tóny. Stupnici lze tedy definovat jako posloupnost intervalů.

Na světě existuje velmi mnoho stupnic. V této práci bude látka omezena na stupnice evropské hudby, které mají v dnešní době význam. Nejznámější stupnice je jónská, kterou známe pod názvem durová, a stupnice aiolská, kterou známe jako stupnici mollovou.

V tabulce 2.4 je seznam používaných stupnic. Sloupec sled intervalů znázorňuje posloupnost intervalů podle tabulky 2.3.

Název	Sled intervalů
Jónská	2, 2, 1, 2, 2, 2, 1
Dórská	2, 1, 2, 2, 2, 1, 2
Frigická	1, 2, 2, 2, 1, 2, 2
Lidická	2, 2, 2, 1, 2, 2, 1
Mixolidická	2, 2, 1, 2, 2, 1, 2
Aiolská	2, 1, 2, 2, 1, 2, 2
Hypofrigická	1, 2, 2, 1, 2, 2, 2

Tabulka 2.4: Výčet druhů stupnic

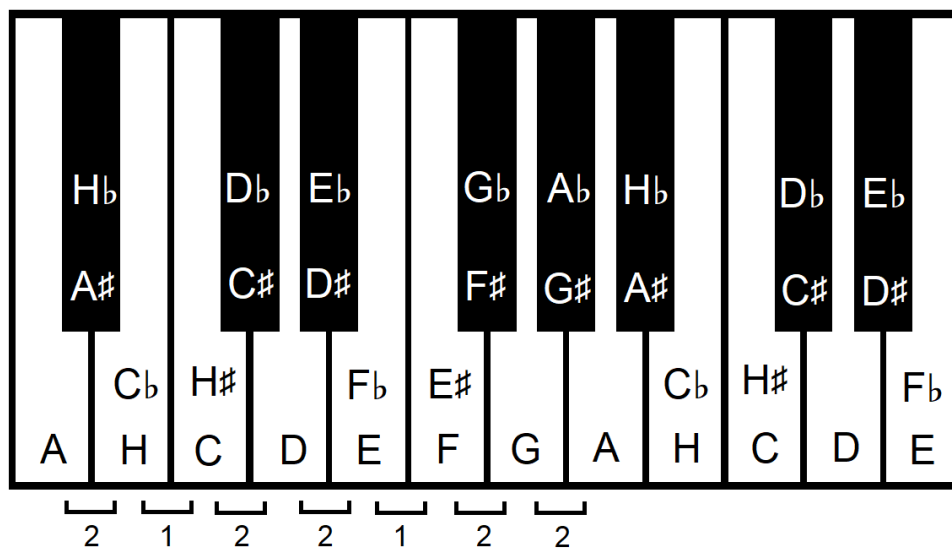


Obrázek 2.2: Stupnice C dur znázorněná na klaviatuře

Každou stupnici lze vytvořit od kteréhokoli tónu. Výchozí tón stupnice je základním tónem stupnice, která se podle něj jmenuje. Například stupnice C dur (viz obrázek 2.2) začíná tónem C, stupnice A moll (viz obrázek 2.3) začíná tónem A.

2.5 Akordy

Akord je souzvuk minimálně tří tónů, které mají různou výšku. Zahráním tónů po sobě vzniká akord rozložený. Zahráním tónů najednou vzniká akord harmonický.



Obrázek 2.3: Stupnice A moll znázorněná na klaviatuře

Akordy se odvozují podle stupnic nebo podle intervalů. Nejčastěji podle počtu tónů, které jsou obsažené v akordu v rozmezí jedné oktávy, se akordy dělí na *trojzvuky*, *čtyřzvuky*, *pětizvuky* a *vícezvuky*.

Obdobně jako u stupnic je v hudbě spousta druhů akordů a v této práci budou zmíněny pouze akordy, které se běžně vyskytují v evropské hudbě. V tabulce 2.5 jsou základní typy akordů a odvození je pro zjednodušení uvedené pomocí posloupnosti intervalů, podobně jako je tomu u stupnic³.

Obdobně jako u stupnic lze utvořit akord od kteréhokoli tónu. Výchozí tón akordu je základním tónem akordu a dává mu jméno.

³Odvození podle stupnic je velice náročné, například akord *C7* obsahuje 0., 2., 4. a zmenšený 6. tón ve stupnici *C* dur.

Název	Souzvuk	Sled intervalů
dur	trojzvuk	4, 3
moll	trojzvuk	3, 4
7	čtyřzvuk	4, 3, 3
maj7	čtyřzvuk	4, 3, 4
6	čtyřzvuk	4, 3, 2
add9	čtyřzvuk	4, 3, 7
sus4	trojzvuk	5, 2
dim	trojzvuk	3, 3

Tabulka 2.5: Výčet druhů akordů

Trénování hudebního sluchu

Trénovat můžeme absolutní nebo relativní sluch. Mít absolutní sluch znamená, že bychom dokázali určit přesnou výšku tónu bez dřívějšího poslechu nebo pamatování nějakého referenčního tónu. Mít relativní sluch znamená, že dokážeme určit výšku tónu vzhledem k nějakému referenčnímu tónu. [4]

V následujících kapitolách budou uvedeny elementární disciplíny pro trénink hudebního sluchu. Ve spolupráci s pedagogem Františkem Pelíškem [3] bude představena možná strategie pro trénink hudebního sluchu.

3.1 Elementární disciplíny

Hudební sluch lze trénovat v několika dílčích oblastech. Můžeme trénovat slyšitelný rozsah tónů. Podle knihy [2] může lidské ucho vnímat tóny s frekvencí v intervalu přibližně od 20 Hz do 20 000 Hz. Můžeme trénovat oba extrémy, tedy trénovat poslouchání tónu s velmi nízkou frekvencí a naopak s velmi vysokou frekvencí, a tím si rozšířit rozsah intervalu.

Další možností, která se nabízí, je postupně se naučit porovnat výšky dvou tónů, tedy poznat, který z tónů je vyšší a který nižší. Pokud bychom dokázali porovnat tóny s velmi malým rozdílem jejich frekvencí, dokázali bychom si s přehledem naladit hudební nástroj.

Obdobně jako rozeznávání výšky tónů bychom mohli cvičit odlišování síly tónů. Při nácvičování rozlišování barvy tónu bychom postupem času dokázali rozeznat jednotlivé hudební nástroje.

3.2 Strategie pro trénování hudebního sluchu

Pro trénování hudebního sluchu neexistuje zaručená praktika. Nezbyvá tedy nic jiného než trénink uskutečnit v podobně opakovaného přehrávání tónů a natrénovat si na nich jejich vlastnosti, které jsou uvedeny v kapitole 2.1.

Podle studií a výzkumů uvedené v diplomové práci [4] je výcvik pravého absolutního sluchu velmi náročný a jsou zapotřebí některé predispozice. Snaha o naučení nepravého absolutního sluchu je ovšem možná. Při trénování absolutního sluchu bychom postupem času mohli umět klasifikovat výšky tónů do oktáv. Opakovaným tréninkem bychom pak mohli zpřesňovat polohu, až bychom dokázali identifikovat tón.

Pro zkvalitnění relativního sluchu bychom měli začít se základními intervaly (*prima* a *oktáva*) v základním oktávovém systému – jednočárkované oktávě. Postupem času se přidají intervaly *kvarta* a *kvinta*, v další iteraci *sekunda*, *tercie* a zbytek základních intervalů (*sexta* a *septima*). Poté bychom mohli zdokonalovat intervaly – rozšířené. Další modifikace by spočívala v přehrávání intervalů v opačném pořadí tónů – první tón by byl vyšší a druhý nižší. Poté by se prodleva mezi zahranými tóny mohla snižovat, až by se tóny zahrály najednou. Nakonec bychom měli trénovat vše dohromady v různých oktávových systémech a snažit se trénovat do extrémů – intervaly ve vysokých a nízkých polohách.

Po úspěšném zdolání intervalů bychom měli pokračovat s určováním rozdílů výšek dvou tónů. Uživatel by měl být schopný určit, který tón je vyšší s rozdílem o celý tón, poté o půltón, o čtvrt tónu až nakonec o šestnáctinu tónu.

Rozšířením intervalů dostáváme nové disciplíny – stupnice a akordy. Stupnice i akordy se odvozují z intervalů podle předepsaných pravidel a je zbytečné se je učit bez předchozího naučení intervalů.

U stupnic bychom měli začít s durovými a mollovými stupnicemi. Dále můžeme pokračovat v libovolném pořadí ostatních stupnic uvedené v tabulce 2.4. U akordů bychom měli jako u stupnic začít s durovými a mollovými akordy. Poté bychom měli umět rozeznat akordy, které se liší přidáním 1 tónu od durového nebo mollového akordu (7, maj7, 6, add9). Z tabulky 2.5 zůstávají akordy sus4 a dim.

Nakonec se můžeme vyvíjet v hudebním cítění – trénovat barvu (rozpoznání hudebních nástrojů) a délku (rytmus).

Analýza

V předchozí kapitole byly popsány elementární disciplíny a strategie pro trénink hudebního sluchu. V této kapitole se naváže na návrh funkčních a nefunkčních požadavků aplikace, které by umožnily efektivně trénovat hudební sluch podle uvedené strategie. V neposlední řadě se v této kapitole porovná a zhodnotí konkurenční řešení vůči požadavkům aplikace.

4.1 Funkční a nefunkční požadavky

Aplikace by měla jednoduše umožnit naplánovat tréninky v podobě jednotlivých dílčích úloh, ve kterých by uživatel poznával daný zvuk, a měla by umožnit trénovat základní disciplíny hudebního sluchu (viz kapitola 3). Měla by klást důraz na podrobnou statistiku v průběhu celého tréninku a zaznamenávat například počet přehrání daného zvuku nebo poměr mezi správnými a špatnými odpověďmi. Z této statistiky by poté mohla aplikace zjistit slabé a silné stránky uživatele a na základě toho by vygenerovala další tréninky.

Dále by aplikace měla umožnit konfigurovatelnost katalogů pro přehrávání intervalů, stupnic a akordů. Přehrát bychom je mohli v libovolné oktávě, pro libovolný počáteční tón, pro různé hudební nástroje a různou prodlevu mezi zahranými tóny.

V neposlední řadě by měla aplikace motivovat uživatele ke zlepšování svých dovedností pomocí *Learn through play* aktivit, mezi které patří například získávání různých odměn, získávání virtuálních bodíků, různá vylepšení apod. K tomu je zapotřebí vlastnit účet. Uživatel by měl mít možnost si účet založit, upravit nebo zrušit.

Do funkčních požadavků se v této práci nezahrne disciplína pro trénink hlasitosti tónů, jelikož by příliš hlasitými tóny mohla aplikace poškodit zdraví uživatele. Naopak u příliš slabých tónů se nedá zaručit, že uživatele nevyruší vnější prostředí. Rytmus do funkčních požadavků také nebude zahrnutý, protože spíš souvisí s hudebním cítěním.

Kromě těchto funkčních požadavků by měla aplikace běžet i bez připojení k internetu. Aplikace by měla fungovat na operačním systému Android nebo iOS, které patří v dnešní době mezi nejrozšířenější operační systémy pro mobilní telefony. Dále by měla být jednoduchá v ovládní a mít přehledné uživatelské rozhraní. V neposlední řadě by měla být navržena tak, aby se dala jednoduše rozšířit.

Seznam funkčních a nefunkčních požadavků aplikace:

F1 Správa účtů

F2 Snadné plánování tréninků

F3 Trénování intervalů

F4 Trénování porovnání výšek dvou tónů

F5 Trénování akordů a stupnic

F6 Trénování hudebních nástrojů

F7 Přehled o podaném výkonu v podobě podrobných statistik

F8 Nastavitelný přehled intervalů, stupnic, akordů

F9 Motivace v podobě získávání různých odměn a dohánění soupeřů

N1 Aplikace musí běžet bez připojení k internetu

N2 Aplikace musí fungovat na operačním systému Android nebo iOS

N3 Jednoduchost ovládní a přehledné uživatelské rozhraní

N4 Rozšiřitelnost v oblasti přidávání nových disciplín (např. hudební cítění)

4.2 Vyhodnocení konkurenčních aplikací

Mezi konkurenční aplikace jsem zařadil aplikace s velmi pozitivním hodnocením, které splňovaly většinu z funkčních požadavků a neměly navzájem podobné strategie trénování. Vybral jsem mezi ně aplikace The Ear Gym [5] a Perfect Ear [6]. Pro srovnání ještě zvolil aplikaci Functional Ear Trainer [7], která trénuje hudební sluch zcela jinou metodou.

4.2.1 The Ear Gym

Z požadovaných funkcionalit nabízí aplikace The Ear Gym (viz obrázek 4.1) trénování intervalů, akordů a pro uživatele s prémiovým účtem i trénink stupnic. Nabízí i řadu dalších možností, které souvisejí spíš s hudební naukou. Patří mezi ně čtení a sestavování intervalů a jejich inverzí podle notového zápisu. Kromě toho se zaměřuje také na trénink zpěvu.

Aplikace funguje na bázi „řekni mi, co bys rád trénoval, a já ti to budu přehrávat“ – Pokud chceme trénovat některou z uvedených disciplín, musíme aplikaci říct, které konkrétní intervaly, akordy nebo stupnice chceme trénovat. Po konfiguraci se spustí test, který ve smyčce zadává úkoly. V každém kole se přečte správná nebo špatná odpověď. Pokud odpovíme chybně, nabízí se nám přehrání špatně zvolené odpovědi, kterou můžeme porovnat se správným řešením. Ukončení testu se provede tlačítkem zpět. Poté se nám objeví statistika.

Trénování intervalů probíhá dvěma způsoby. Můžeme trénovat identifikaci jednotlivých intervalů nebo můžeme porovnávat dva intervaly a určit, který z nich je větší.

Mezi nedostatky aplikace patří nesourodost testů, které se odehrávají pouze ve 2–3 oktávách. Prodleva mezi zahranými tóny se nemění. Zásadním problémem u akordů je, že už od počátku se akord zahraje harmonicky najednou a není možnost přehrání akordu v melodickém rozkladu.

Další nedostatek je nepřítomnost průvodce. Uživatel si musí předem nakonfigurovat test, a stává se tedy sám učitelem, což může být dobré pro uživatele s hudebním sluchem, kteří by si rádi potrénovali konkrétní zvuky. Ovšem pro úplně začátečníky to může být naprosto neefektivní způsob, protože sami neví, čím by měli začít.

Aplikace není vhodná pro začátečníky. S aplikací se spokojí uživatel, který potřebuje trénovat konkrétní interval nebo akord, ovšem nic dalšího mu aplikace nenabídne. Na Google Play (viz [5]) jsou o aplikaci dále dostupné následující informace:

Počet stažení přes 50 000

Výsledné hodnocení 4,7/5

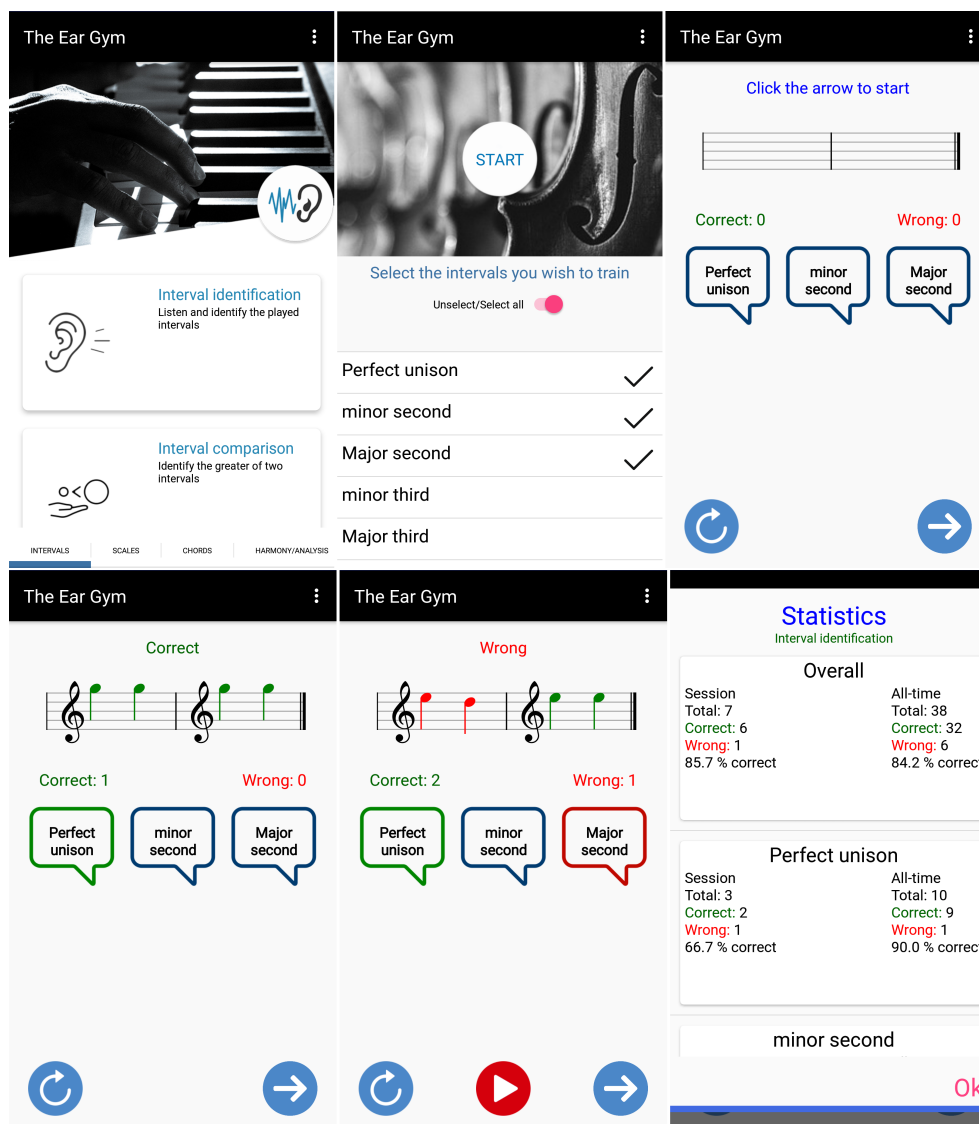
Počet hodnocení a recenzí přes 1 000

Datum vydání 12. 9. 2017

4.2.2 Functional Ear Trainer

Aplikace Functional Ear Trainer (viz obrázek 4.2) je navržena podle Alaina Benbassata, který přišel s novou, revoluční metodikou pro trénink hudebního sluchu. Aplikace neumožňuje trénink jednotlivých disciplín, nýbrž trénuje sluch komplexně. Přehrává tzv. kadence stupnice, což je posloupnost akordů

4. ANALÝZA



Obrázek 4.1: Ukázky aplikace The Ear Gym [5, snímek pořídil autor]

podle nějakého pravidla (např. kadence ke stupnici *C* dur jsou akordy *C* dur, *F* dur, *G* dur, *C* dur) a k tomu zahraje tón ze stupnice *C* dur, který máme za úkol určit.

Aplikace nesplňuje mnoho z uvedených požadavků. Neumožňuje měnit dobu mezi přehranými tóny. Obsahuje doporučený průchod trénování a nastavení vlastního tréninku, ale chybí průvodce. Jako pozitivní hodnotím podrobnou statistiku.

Vzhledem k tomu, že tato metodika netrénuje sluch od základů, nýbrž komplexně, není vhodná pro úplné začátečníky. Pro zdatnější muzikanty by to mohla být efektivní alternativní metodika. Na Google Play (viz [7]) jsou o aplikaci dále dostupné následující informace:

Počet stažení přes 500 000

Výsledné hodnocení 4,8/5

Počet hodnocení a recenzí přes 8 000

Datum vydání 13. 11. 2017

4.2.3 Perfect Ear

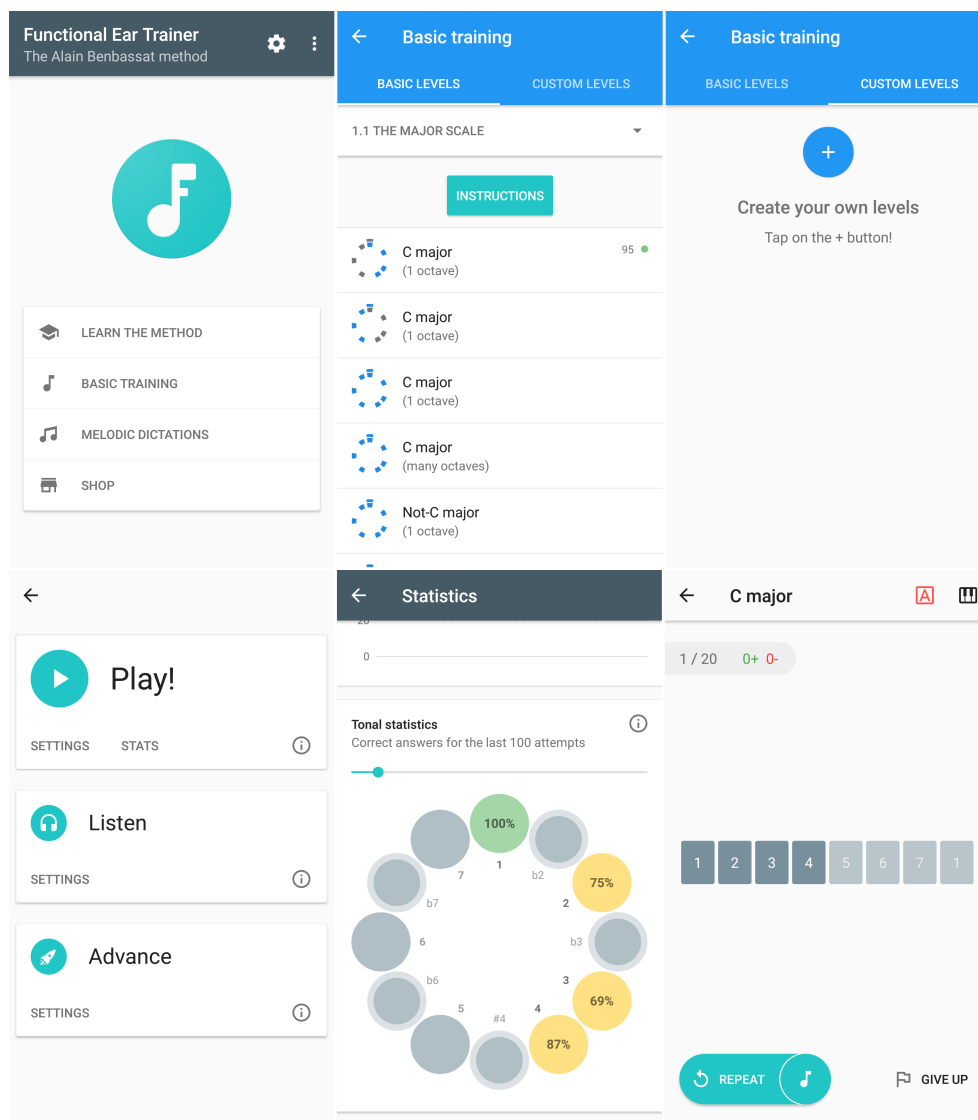
Z uvedených konkurenčních aplikací je Perfect Ear (viz obrázek 4.3) nejpracovanější a splňuje nejvíce z uvedených požadavků. Nabízí trénink intervalů, akordů a stupnic. Mimo jiné umožňuje trénink absolutního sluchu, zpěvu a rytmu. Aplikace také jako jediná z uvedených obsahuje přehled všech intervalů, akordů a stupnic, které si může uživatel přehrávat, aniž by spouštěl nějaký test.

Pro jednotlivé disciplíny obsahuje aplikace doporučený průchod, ale uživatel si může nadefinovat vlastní úroveň. Doporučený průchod se ovšem liší od strategie pro trénink hudebního sluchu, který jsem popsal. Průvodce zde chybí, ale na základě podrobné statistiky nám aplikace vnutí možnost si vylepšit statistiku u jednotlivých disciplín.

Trénink intervalů probíhá dvěma způsoby, jako tomu je v aplikaci The Ear Gym. Hlavní výhodou aplikace je její motivační část. Aplikace nabízí řadu *Learn through play* aktivit. Každá zdolaná úroveň je ohodnocena body za celkovou úspěšnost. Nabízí další bonusové body (časový bonus) a mínusové body (přeskočení úkolu) a další mechanismy na vylepšení, jako například získání achievementů (splnění úkolů s nejlepším hodnocením, trénování několik dní po sobě bez vynechání atd.). Uživatel může využít propojení s účtem Google. Snaha překonat soupeře může působit motivačně.

Mezi nedostatky aplikace patří nesourodost testů. Prodleva mezi zahranými tóny se nemění. Dalším slabým místem je omezený přehled intervalů a akordů. Dovoluje pouze přehrání vzestupně, sestupně a najednou. Chybí

4. ANALÝZA



Obrázek 4.2: Ukázky aplikace Functional Ear Trainer [7, snímek pořídil autor]

4.2. Vyhodnocení konkurenčních aplikací

Požadavky	The Ear Gym	Functional Ear Trainer	Perfect Ear
Intervaly	ano	nepřímo	ano
Porovnání výšek	ne	ne	ne
Stupnice	pro prémiové uživatele	ano	ano
Hudební nástroje	ne	dá se změnit neprovádí testy	dá se změnit neprovádí testy
Hlasitost	ne	ne	ne
Způsob trénování	vlastní nedefinované testy	doporučený průchod	doporučený průchod
Vlastní trénink	ano	ano	ano
Katalog	ne	ne	není dostatečně škálovatelný
Statistika	není dlouhodobá	ano	ano
Learn through play	soutěžení	ne	motivační odměny (body, achievements) soutěžení (propojení s Google)

Tabulka 4.1: Shrnutí konkurenčních řešení

zde škálovatelnost polohy počátečního tónu, a tím nedocílíme trénink hudebního sluchu v nízkých a vysokých polohách. Na Google Play (viz [6]) jsou o aplikaci dále dostupné následující informace:

Počet stažení přes 1 milion

Výsledné hodnocení 4,8/5

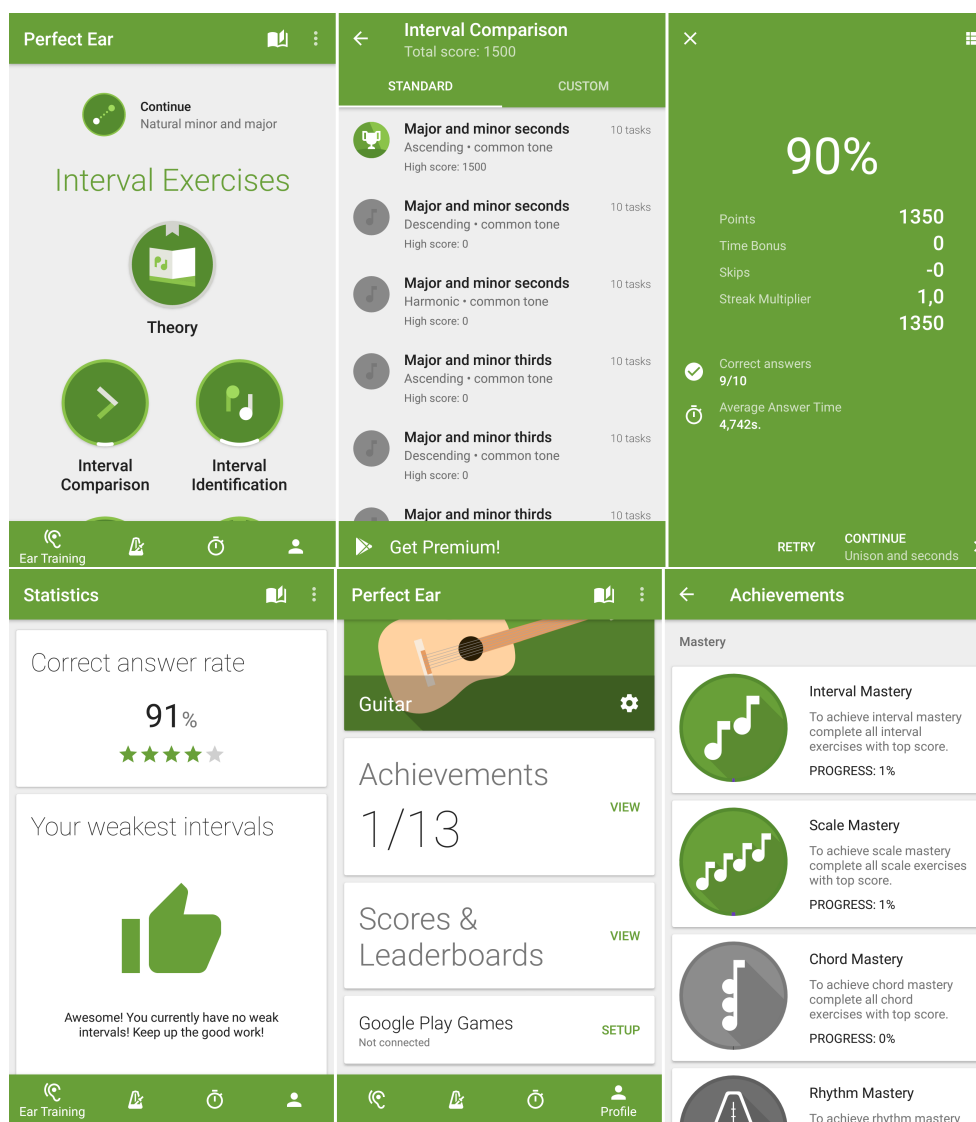
Počet hodnocení a recenzí přes 55 000

Datum vydání 25. 10. 2013

4.2.4 Shrnutí

Z tabulky 4.1 vidíme, že aplikace Perfect Ear se velice přibližuje požadovanému řešení. Mezi silné stránky patří bohatý program *Learn through play*. Slabé stránky aplikace tvoří nedostatečně škálovatelný katalog, nedostatečně různorodé testy a absence trénování porovnání výšek dvou tónů, které je jednou z elementárních disciplín.

4. ANALÝZA



Obrázek 4.3: Ukázky aplikace Perfect Ear [6, snímek pořídil autor]

Návrh řešení

V této kapitole bude nejdříve popsána funkcionalita výsledné aplikace. Na základě porovnání konkurenčních aplikací s funkčními požadavky se zjistilo, že některé funkcionality neodpovídají původním požadavkům nebo zcela chybí. V této části se navrhnu změny, které by více odpovídaly původním požadavkům a budou doplněny chybějící funkcionality.

Následuje návrh samotné tvorby aplikace, která začne v konceptu uživatelského rozhraní, které odpovídá use-case požadavkům. Nakonec se bude kapitola zabývat vhodnou architekturou aplikace a návrhem datového modelu, který pokryje veškerou funkcionalitu.

5.1 Navržené změny oproti konkurenčním aplikacím

Aplikace Perfect Ear nabízí katalog pro přehrávání intervalů, stupnic a akordů. Přehrávání je možné vzestupně, sestupně a harmonicky (souzvuk tónů), chybí ovšem nastavení počátečního tónu a změna prodlevy mezi zahranými tóny. První navrženou změnou je tento přehled zvuků udělat konfigurovatelným. Uživatel by si mohl nakonfigurovat interval, stupnici nebo akord podle své potřeby. Mohl by nastavit přehrané tóny, výšku počátečního tónu, prodlevu mezi zahranými tóny a mohl by si vybrat i hudební nástroj. Podlé mého názoru je konfigurovatelnost katalogů jednou z primárních vlastností, které by aplikace tohoto typu měla nabízet. Uživatel pak nemusí vlastnit žádný hudební nástroj, ani nemusí na žádný hudební nástroj umět hrát a bude si moci přehrát libovolný zvuk.

Další věc, která by se mohla změnit a prostřednictvím níž by se odlišila výsledná aplikace od konkurenčních, je koncept trénování. Uživatel by neměl doporučený průchod, kterého by se držel, ale pomáhal by mu nějaký průvodce. Ten by byl schopný navrhnout testy podle aktuálních schopností uživatele, což bude hlavním cílem mnou vyvíjené aplikace. Průvodce by se snažil tréninky

generovat podle kapitoly 3.2. Uživatel by pak nemusel vymýšlet, co by měl trénovat, a ani by nemusel vědět, které lekce budou následovat.

Zásadním nedostatkem konkurenčních aplikací je absence trénování porovnání výšek dvou tónů, které patří mezi elementární disciplíny trénování hudebního sluchu. Tuto disciplínu bych zavedl do navrhovaného řešení, které by mohlo fungovat následujícím způsobem. Uživateli se přehrají dva tóny, jejichž frekvence se liší o několik tónů. Následně by se podíl zmenšoval, až by konvergoval k poměru 1 : 1. Postupem času by se obtížnost mohla zvyšovat například tím, že by se tóny přehrávaly v jiných výškových polohách. Po absolvování tréninku by v ideálním případě byl uživatel schopen si naladit hudební nástroj a poznat falešně znějící tóny.

Jednotlivé tréninky v konkurenčních aplikacích jsou velmi inspirující, avšak v několika ohledech si zaslouží úpravu. Jednak bych změnil doporučený průchod intervalů, který se u konkurenčních aplikací liší od kapitoly 3.2, a jednak bych upravil nabídku správných odpovědí. Ani jedna aplikace nenabízí možnost „žádná odpověď není správná“, což by mohlo zlepšit trénink. Dalšího zkvalitňování bude docíleno, pokud by se v testech měnily hudební nástroje a uživatel by pak vybíral z více odpovědí. Bonusové body za čas bych zrušil, co nejrychleji zmáčknout správnou odpověď není cílem tréninku. Zavedl bych spíš záporné body za to, že člověk využije opětovné přehrávání. To bych zahrnul i do výsledné statistiky, aby uživatel věděl, kolikrát si musel přehrát daný zvuk, aby byl schopný odpovědět správně.

5.2 Popis funkcionalit aplikace

Pro vstup do aplikace je potřeba uživatelského profilu. K jeho vytvoření si uživatel zvolí unikátní uživatelské jméno a heslo. Dále může vyplnit věk, zemi a pohlaví. Přihlášením se dostane do aplikace. Je možné zvolit trvalé přihlášení. V sekci profil v hlavním menu lze upravit uživatelské údaje nebo účet zrušit.

V sekci „Úkoly“ (Tasks) si uživatel plánuje tréninky, volí pouze den, kdy dostane upozornění přes notifikaci. Dále volí disciplíny, které by chtěl trénovat. Průvodce na základě statistik uživatele vytvoří individuální trénink, při němž si uživatel potrénuje své slabé stránky. Uživatel bude moci trénink uskutečnit, nebo přesunout na jiný den.

V přehledu intervalů uživatel smí volit počáteční tón a jeho tónovou soustavu, hudební nástroj, prodlevu mezi zahranými tóny intervalu a sestupný nebo vzestupný způsob přehrávání. Pro stupnice si uživatel volí počáteční tón, tónovou soustavu, druh stupnice a hudební nástroj. Stupnice se zahrají nejdříve vzestupně a následně sestupně. U akordů uživatel vybírá druh akordu, počáteční tón, tónovou soustavu, hudební nástroj a způsob přehrávání akordu.

Statistika bude uživateli zobrazovat celkové ohodnocení jednotlivých disciplín a následně jejich detailní rozbor. Uživatel si bude moci zobrazit dlouhodobou i trendovou statistiku, na které je závislé generování testů.

V sekci „Žebříček“ (Rank) se bude uživatel moci informovat o aktuálním trendu ve světě. Dozví se, jak často trénují ostatní uživatelé a jak pokročilý je vůči ostatním uživatelům. Uživatel může získávat za splněné testy body, které se započítají do celkového umístění. Žebříček bude umožňovat filtraci podle věku nebo země.

V sekci „Ocenění“ (Achievement) bude souhrn zvláštních ocenění za splnění neobvyklých úkolů, jako je například dokončení disciplíny, nebo za strávení určitého času tréninkem, určení správných odpovědí v řadě apod.

5.3 Návrh uživatelského rozhraní

Na obrázku 5.2 je vlevo nahoře návrh obrazovky pro tréninky v podobě jednotlivých úkolů. Na obrazovce je seznam úkolů, které si uživatel naplánoval. Pomocí horního tlačítka v pravém rohu obrazovky si bude uživatel moci naplánovat nové tréninky. Zobrazí se mu nabídka pro plánování, v níž si uživatel zvolí datum a dostupné disciplíny, které by v daný den chtěl trénovat.

V pravém horním rohu obrázku 5.2 je vyobrazené menu, které se objeví po přejetí prstu z levého kraje obrazovky směrem doprava nebo po kliknutí na tlačítko menu, které je umístěné v levé části záhlaví. Po přejetí prstu v opačném směru, od pravého okraje menu k levému kraji obrazovky, se menu opět schová.

Na obrázku 5.2 vlevo dole vidíme návrh obrazovky průběhu tréninku. Uživatel si bude moci opakovaně přehrávat zvuk, který má za úkol uhádnout. Zvolí odpovědi a potvrdí tlačítkem „Submit“.

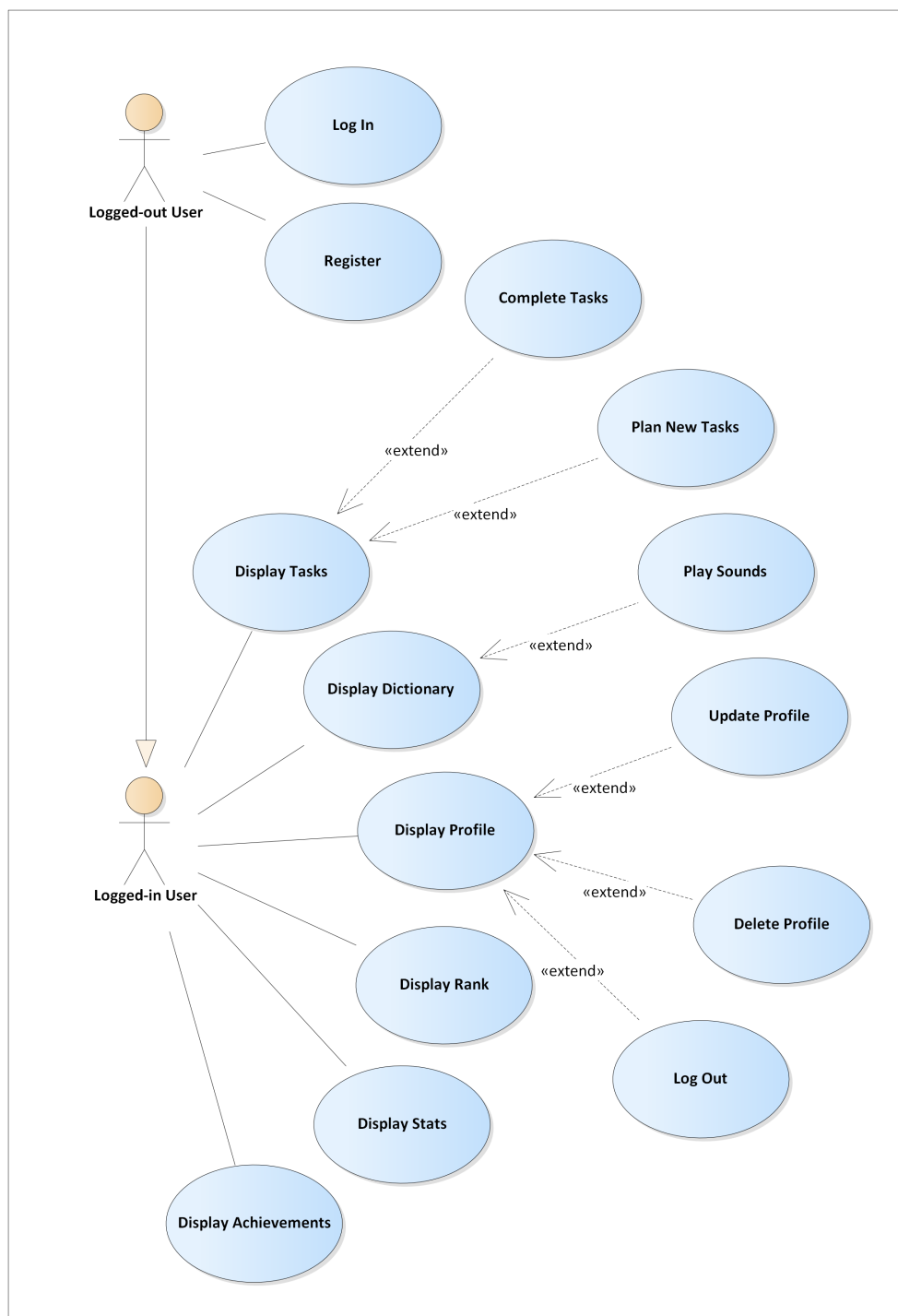
Následuje obrazovka přehledu intervalů, stupnic a akordů. Uživatel si v dolním panelu zvolí disciplínu, nakonfiguruje si daný zvuk a může si zvuk přehrávat.

Poslední obrazovka demonstruje návrh žebříčku, který umožňuje filtrování podle země a věkové kategorie.

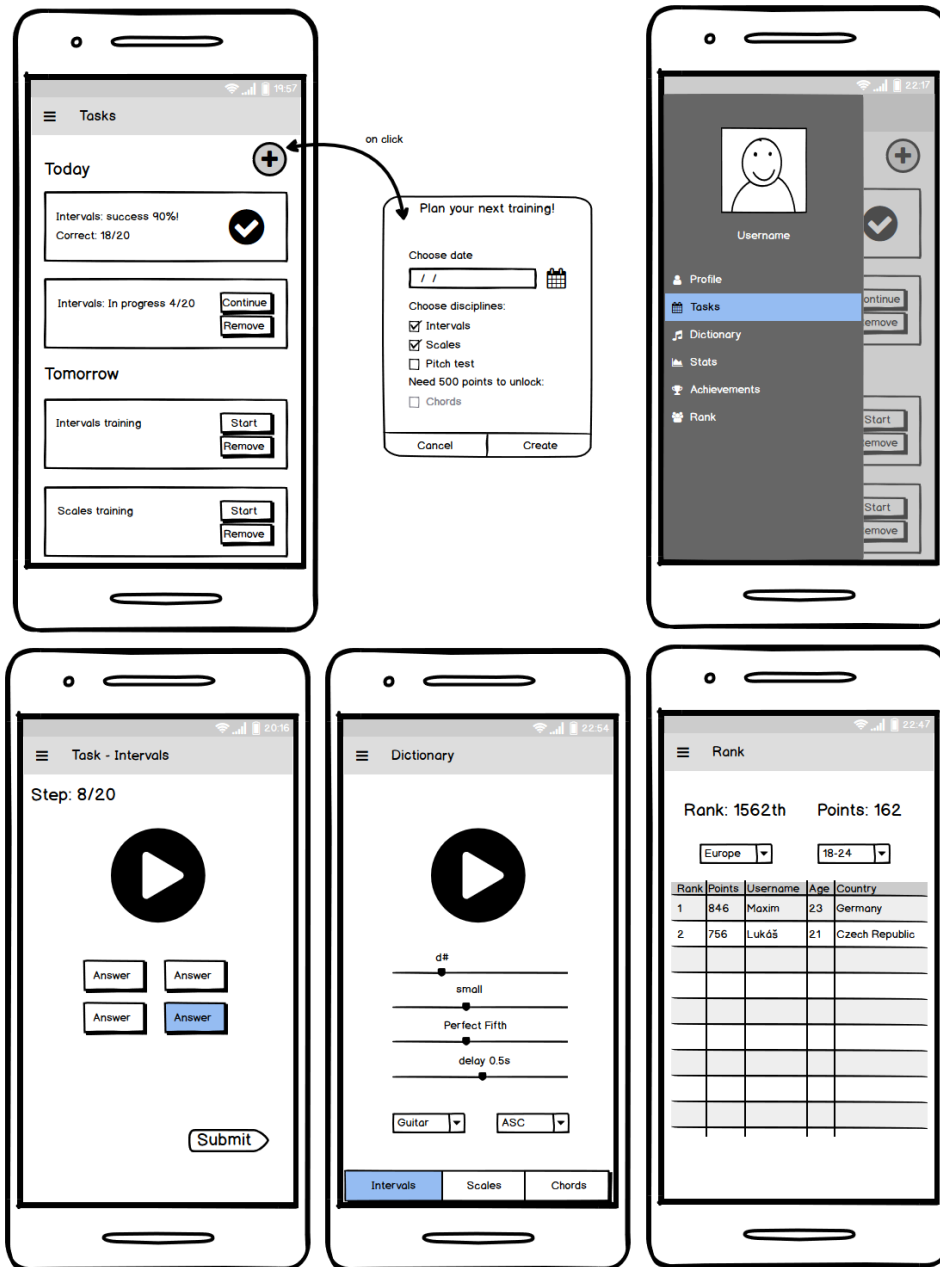
5.4 Architektura

Model-View-ViewModel (zkráceně MVVM) je návrhový vzor pro oddělení prezentační vrstvy od obchodní logiky v uživatelském rozhraní aplikace. Oddělené vrstvy tak značně usnadňují testování například pomocí unit-testů, které nejsou závislé na uživatelském rozhraní. [8, 9]

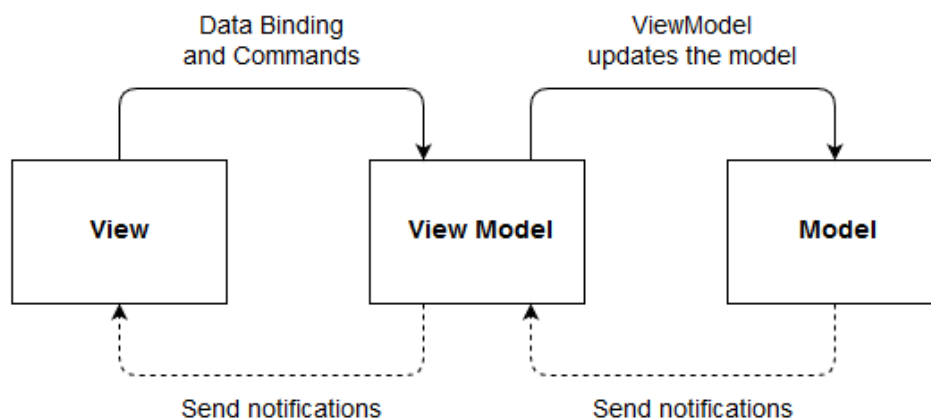
5. NÁVRH ŘEŠENÍ



Obrázek 5.1: Diagram případů užití



Obrázek 5.2: Wireframe aplikace



Obrázek 5.3: Architektura Model-View-ViewModel [10]

MVVM je tak mnohem lepší na údržbu a vývoj aplikací než například klasický návrhový vzor MVC. To může také výrazně zlepšit možnosti opětovného použití kódu a umožňuje vývojářům a návrhářům uživatelského rozhraní snadněji spolupracovat při vývoji příslušných částí aplikace. [8, 9]

MVVM se skládá ze 3 částí [8, 9], které jsou svázané podle obrázku 5.3:

Model obvykle obsahuje datový model spolu s obchodní a validační logikou.

View reprezentuje uživatelské rozhraní.

View model spojuje Model s View a drží si stav aplikace. Upozorňuje View na změny stavu prostřednictvím událostí. Pomocí data binding jsou propojeny ovládací prvky View s ViewModel a čerpají z něj obsah. View-Model upravuje Model, který zpátky upozorňuje ViewModel na změny prostřednictvím událostí.

V našem případě budou v modelu obsaženy služby pro komunikaci s databází, které jsou popsány v následující sekci 5.5, a služba pro spuštění a generování zvuků. Do modelu zahrneme ještě vyhodnocení statistiky společně s vytvářením příslušných tréninků.

View bude reprezentovat uživatelské rozhraní, které bude vypadat jako na obrázku 5.2.

Ve ViewModel budou udržovány aktuální modely, které dohromady reprezentují aktuální stav aplikace. Ve ViewModel budou uloženy aktuální údaje od uživatele, jeho úkoly, jeho statistika apod.

5.5 Databázová vrstva

Databázový model je zachycen na obrázku 5.1. V modelu jsou obsažené katalogové tabulky a produkční tabulky, které jsou rozděleny podle typu obsažených položek.

Typicky jsou v katalogové části tabulky, které představují obecné položky pro aplikaci, které se zpravidla během používání aplikace nemění a také nejsou během používání aplikace přidávány nebo odebírány. Katalogovými tabulkami jsou:

Country katalog států

Achievement katalog odměn

Discipline katalog disciplín hudebního sluchu

Tone katalog tónů v základní tónové řadě

Octave katalog oktáv

Inverval katalog intervalů

Scale katalog stupnic

Chord katalog akordů

Během používání aplikace jsou některé záznamy přidány, odebrány nebo upravovány. Tyto záznamy se zpracovávají v produkčních tabulkách. V našem modelu to jsou tabulky:

User základní údaje o uživateli

UserAchievement získané achievementy uživatele

UserStats uživatelské statistiky

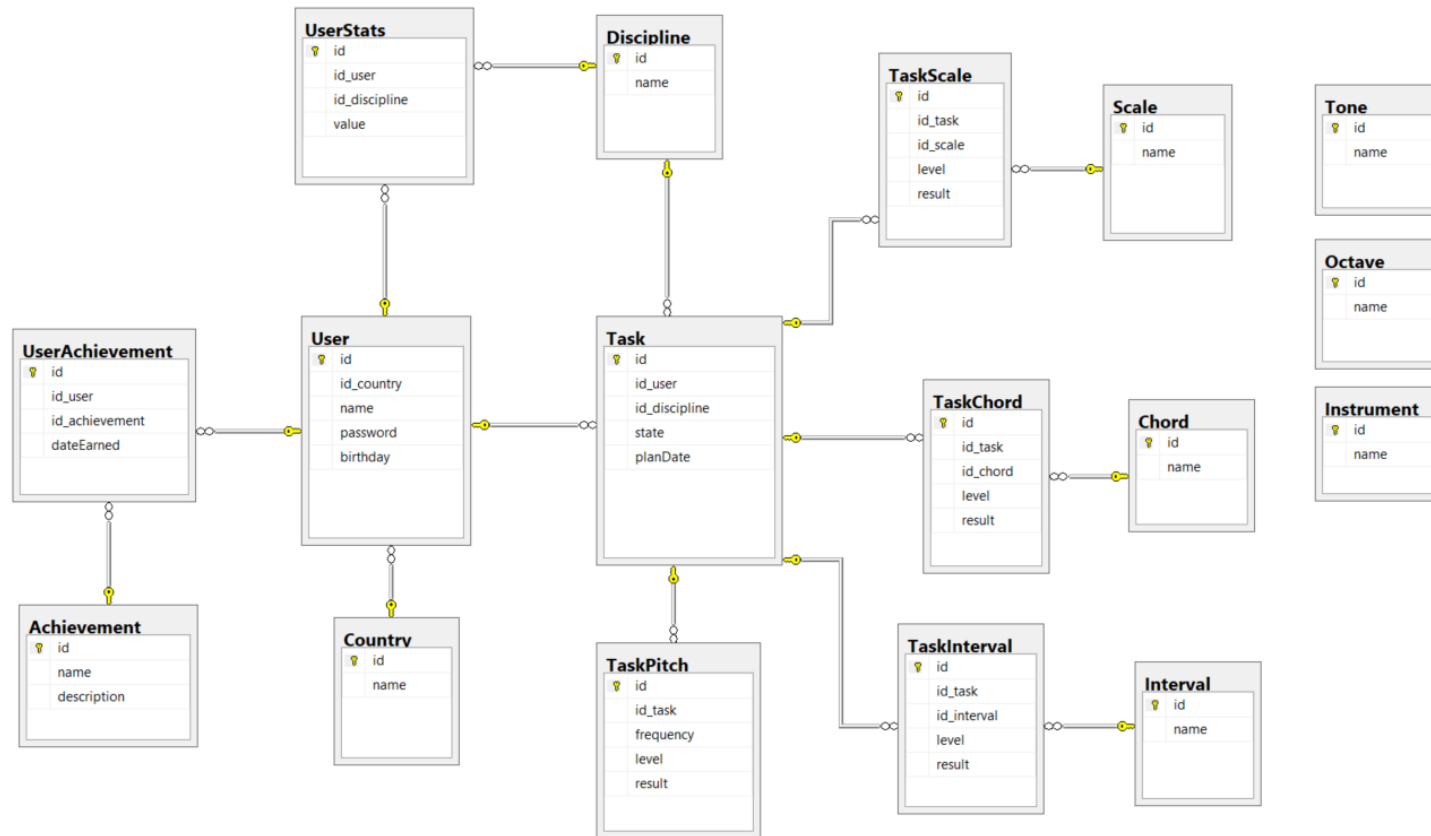
Task úkoly přidělené uživateli

TaskPitch dílčí úkoly v rozpoznání výšek tónů

TaskInterval dílčí úkoly na intervaly

TaskScale dílčí úkoly na stupnice

TaskChord dílčí úkoly na akordy



Obrázek 5.4: Databázová vrstva

5.6 Obchodní logika

Do obchodní logiky aplikace se zahrne veškerá funkcionalita pro ukládání dat, plánování a generování dílčích úkolů v tréninku, vyhodnocení statistiky, spouštění zvuků a dynamické zobrazování uživatelského rozhraní. Tyto funkcionality se rozdělí do určitých rozhraní (interface) a naimplementují se:

Repository Service definuje řadu funkcí pro ukládání a získávání dat uživatele, jeho statistik a tréninků. Umožňuje inicializaci katalogových tabulek a jejich obsah.

Guide Service definuje řadu funkcí pro určení statistik uživatele, od kterých se vyvíjí generování jednotlivých dílčích úkolů v tréninku.

Audio Service definuje řadu funkcí pro přehrávání tónů, intervalů, stupnic a akordů. Umožňuje přehrání zvuků různých hudebních nástrojů a měnit odezvu mezi tóny.

Page Service umožňuje řadu funkcí pro přepínání oken uživatelského rozhraní a pro zobrazení vyskakovacích oken ViewModelům.

Implementace

V této kapitole budou popsány použité nástroje pro vytvoření prototypové aplikace, která byla vyvíjena na operačním systému Windows 10 od společnosti Microsoft. Z toho důvodu je prototyp cílený pouze pro Android.

Dále budou popsány a vysvětleny implementační detaily pro spouštění zvuků, jejich ukládání a možné generování pomocí Fourierovy transformace.

6.1 Nástroje

6.1.1 Xamarin a Xamarin.Forms

Prototypová aplikace byla vytvořena pomocí frameworku Xamarin.Forms, který se liší od základního nástroje Xamarin několika rozšířeními.

Xamarin se používá především pro tvorbu mobilních aplikací a umožňuje současně vyvíjet aplikace pro tři hlavní platformy – Android, iOS a Windows. Xamarin.Forms navíc umožňuje rychlou tvorbu GUI pomocí sdíleného kódu. Obsahuje přes 30 komponent a je velice vhodný pro vytváření prototypových aplikací. Sdílený kód se píše v programovacím jazyce C# a pro popis GUI se používá značkovací jazyk XAML. [11]

6.1.2 Vývojové prostředí

Pro vývoj s Xamarin máme na výběr dvě možnosti. V operačním systému Windows můžeme vyvíjet pouze pro Android a Windows. Je možné používat vývojové prostředí Visual Studio od společnosti Microsoft. [12]

V operačním systému iOS pak můžeme vyvíjet pouze pro Android a iOS. Vývojovým prostředím pro operační systém iOS je primárně Xamarin Studio, ale je možné vyvíjet ve Visual Studio ve virtuálním stroji. [13]

6.1.3 Audacity

Audacity je jednoduchý multiplatformní editor digitálního zvuku. Kromě editace umožňuje také nahrávání. Jedná se o open-source projekt, který je v současnosti vyvíjen skupinou dobrovolníků. [14]

6.1.4 Použité pluginy

Z balíčkovacího správce NuGet jsem pro potřeby ukládání dat využil plugin SQLite. Pro vizualizaci jsem použil Micro-charts pro zobrazení jednoduchých grafů a Input-kit pro užitečné komponenty uživatelského rozhraní. Pro spouštění zvuků jsem využil plugin Simple Audio Player.

NuGet package manager

NuGet je správce balíčků pro .NET ekosystém a hlavní způsob, jak mohou vývojáři vyhledávat a získávat open-source knihovny. NuGet.org je bezplatná webová služba poskytovaná společností Microsoft. Primárně hostují hostitele pro veřejné balíčky NuGet. [15]

SQLite

SQLite je vestavný databázový engine. Na rozdíl od ostatních SQL databází nemá SQLite databázový proces. SQLite čte a zapisuje data přímo do diskových souborů. Kompletní SQL databáze s více tabulkami, indexy, triggers a views je obsažena v jediném diskovém souboru. Formát databázového souboru je multiplatformový. Díky těmto funkcím je SQLite populární volbou jako formát aplikačního souboru. [16]

MicroCharts

MicroCharts je knihovna pro vytváření jednoduchých grafů. Obsahuje řadu grafů, které jsou použity v prototypu aplikace. MicroCharts nabízí Radar-Chart, který se hodí pro znázornění celkových statistik napříč všemi disciplínami. Dále nabízí známý BarChart, který se dá využít pro znázornění detailnější statistiky konkrétní disciplíny. [17]

Input-kit

Input-kit je knihovna pro vytváření jednoduchých komponent sloužících ke zpracování vstupů od uživatele. Mohou se velice prakticky využít v obrazovce pro přihlášení (Advanced Entry), v konfiguraci pro generování zvuků (Slider, Labeled Slider) a v plnění úkolů (CheckBox, Radio Button, Select-onView). [18]

Simple Audio Player

Simple Audio Player je knihovna pro zpracování zvuků pro Xamarin. Umí načíst data ve formátu mp3 a wav. Knihovna umožňuje přehrávat načtené zvuky a pozastavit je při běhu. Umí zároveň pouštět zvuky paralelně. [19]

6.2 Implementace Audio Service

Ukládání jednotlivých zvukových stop je možné například ve formátu wav nebo mp3. Zvukové stopy ve formátu wav jsou mnohem kvalitnější a rychlejší na kódování nebo dekodování, ovšem zabírají více místa. [20]

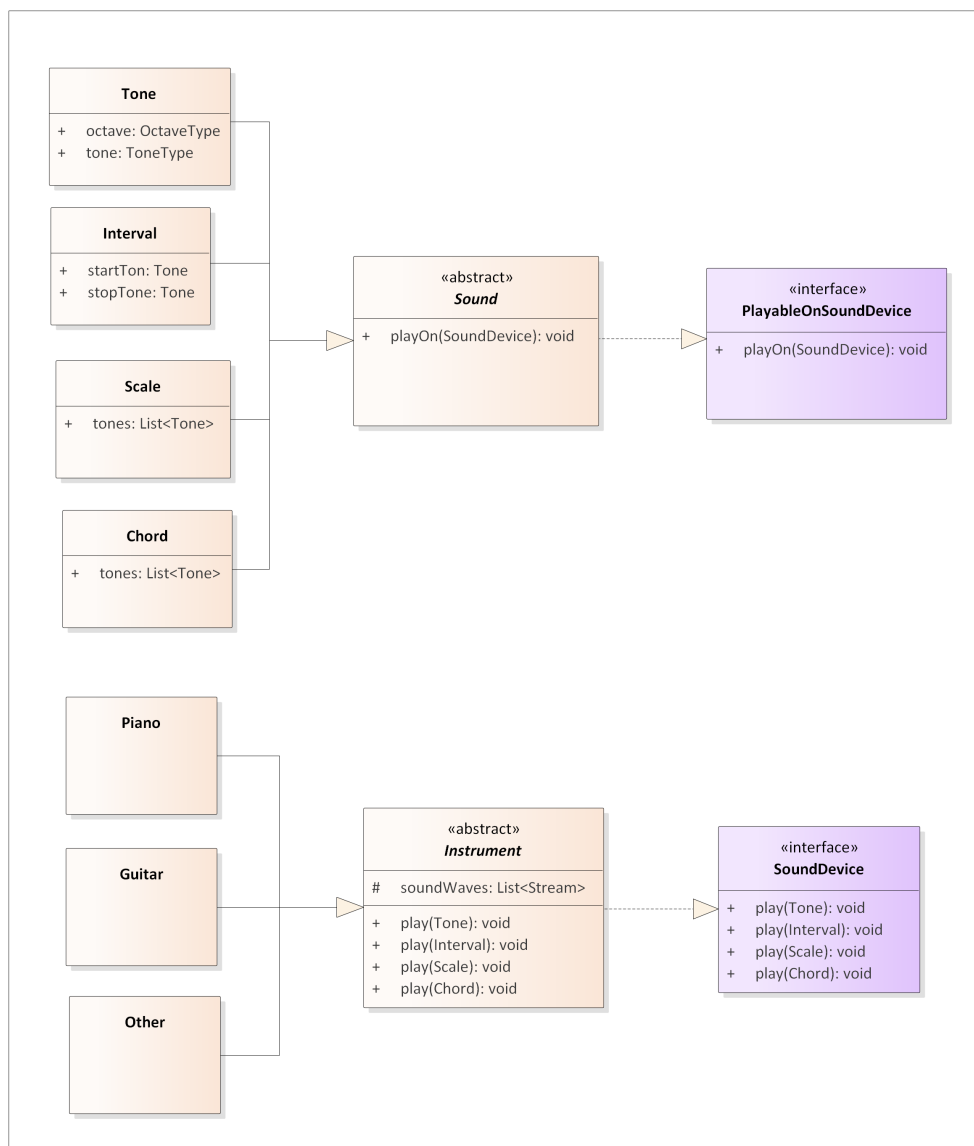
Pro lepší dojem na poslech nahrávek byl zvolen formát wav. Jedna 4s nahrávka zabere přibližně 300 kB. Jedna oktáva na 1 hudební nástroj celkem zabere až 4 MB. V aplikaci byly použity 3 oktávy pro 2 hudební nástroje, pro klavír a kytaru. Celkem zabírají nahrávky kolem 24 MB.

Pro úsporu dat existuje metoda pro generování tónů v jiných frekvencích zvaná Pitch Shift, která využívá Fourierovy transformace. Tímto tématem se zabývala bakalářská práce Jana Onderky [21]. Nevýhoda této metody je, že výstupní zvuky zní velmi počítačově a nepodařilo se odladit algoritmus tak, aby zvuky zněly příjemně na poslech. Z toho důvodu byly nahrávky pořízeny. Nahrávky klavíru jsou dostupné online (viz [22]), nahrávky kytary poskytl student Do Duc Huy [23].

Nahrávky byly počítačovým softwarem Audacity (viz 6.1.3) zkráceny na 4 sekundy a byl na nich aplikovaný efekt Fade Out, který utlumuje nahrávku do ztracena, aby přechod mezi zahranými zvuky nebyl ostrý.

Audio Service by měl umět přehrát tón, interval, stupnici a akordy. Interval se dá reprezentovat jako počáteční a koncový tón s odezvou mezi zahranými tóny. Stupnice a akordy jsou reprezentované jako posloupnosti tónů (viz diagram tříd 6.1). Tyto zvuky by měl umět přehrát Audio Service pro více hudebních nástrojů a měl by být jednoduchý pro rozšíření o další hudební nástroje. K tomuto problému se hodí návrhový vzor Visitor (viz zdrojové kódy 1 a 2).

6. IMPLEMENTACE



Obrázek 6.1: Visitor pattern – Diagram tříd

```

1 public abstract class Instrument : ISoundDevice
2 {
3     // visiting interval - visiting startTone and stopTone
4     public void Play(Interval interval)
5     {
6         Play(interval.StartTone);
7         Play(interval.StopTone);
8     }
9
10    // visiting scale - visiting tones
11    public void Play(Scale scale)
12    {
13        foreach (Tone tone in scale.Tones)
14        {
15            Play(tone);
16        }
17    }
18
19    // visiting chord - visiting tones
20    public void Play(Chord chord)
21    {
22        foreach (Tone tone in chord.Tones)
23        {
24            Play(tone);
25        }
26    }
27
28    // visiting tone
29    public void Play(Tone tone)
30    {
31        Stream soundWave = GetSoundWave(tone.CodeName);
32        Play(soundWave);
33        Delay(tone.Delay);
34    }
35
36    private void Play(Stream soundWave)
37    {
38        ISimpleAudioPlayer player = GetAvailablePlayer();
39        player.Load(soundWave);
40        player.Play();
41    }
42 }

```

Zdrojový kód 1: Visitor patter v C# – navštěvování visitorem

```

1 public abstract class Sound : IPlayableOnSoundDevice
2 {
3     // accept visitor
4     public void playOn(ISoundDevice soundDevice)
5     {
6         soundDevice.play(this);
7     }
8 }

```

Zdrojový kód 2: Visitor pattern v C# – akceptace visitora

Testování

Poslední část této práce se bude zabývat testováním výsledné prototypové mobilní aplikace. Nejdřív budou popsány testy pro správu uživatelských účtů, následně budou popsány testy pro plánování tréninků a nakonec testy pro konfigurovatelnost katalogů pro přehrání intervalů, stupnic a akordů.

Způsob testování a testovací prostředí

Testování probíhalo pomocí interních testů během vývoje a pomocí uživatelských testů. Testování a ladění proběhlo na android emulátoru s verzí operačního systému Android 8.1 Oreo a na reálném mobilním zařízení Samsung Galaxy S5 Neo s verzí operačního systému Android 6.0.1 Marshmallow.

Uživatelských testů se zúčastnili běžní uživatelé operačního systému Android i iOS. Účastníci byli technického i netechnického zaměření, přičemž byla snaha pokrýt cílovou skupinu uživatelů.

Testování správy uživatelských účtů

V testování správy uživatelských účtů byly obsaženy testy pro registraci, přihlášení, úpravu osobních údajů a zrušení účtu. Interní testy se zaměřily na validaci duplicitních nebo nevhodných jmen a nevhodných hesel, poté se zaměřily na databázovou vrstvu. Po registraci se úspěšně uložily potřebné údaje do databáze. Zrušením účtu se otestovalo, jestli byly správně smazané veškeré údaje od uživatele, včetně produkčních záznamů. Interní testy proběhly bez problémů.

Uživatelské testy proběhly pozitivně. Jediný menší problém byl v obrazovce pro správu uživatelských údajů. Uživatel se měl odhlásit a tlačítko na odhlášení se zobrazilo až po posunutí dolů (scroll down). Uživatel byl ovšem schopný po chvilce pátrání dojít k cíli.

Testování plánování tréninků

Další testování probíhalo pro plánování jednotlivých tréninků. Byla ošetřena a otestována funkcionalita, která neumožnila naplánovat tréninky v dny, které již proběhly. Pokud uživatel nesplnil trénink v daný den, přesunul se mu automaticky na aktuální den s varovnou hláškou. Uživatel tak může trénink dokončit nebo zrušit. Interní testy proběhly bez závad.

Uživatelské testy proběhly pozitivně. Jediný menší problém byl v mazání tréninků, které nebylo dostatečně intuitivní především pro uživatele iOS. Mazání tréninku spočívá v dlouhém stisku (long press), po kterém se v horním panelu zobrazí tlačítko na smazání (delete).

Testování Audio Service

Byly provedeny testy pro přehrávání intervalů, stupnic a akordů, pro všechny možné kombinace parametrů, které Audio Service poskytuje. Měnily se přitom různé počáteční tóny, měnil se oktávový systém, prodleva mezi zahranými tóny a změna hudebních nástrojů. Audio Service vyhazuje výjimku, pokud potřebný tón nebyl k nalezení. Tato výjimka se odchytává ve ViewModelu, který zobrazí na uživatelském rozhraní vyskakovací okno s varovnou zprávou.

Závěr

Cílem práce bylo vytvořit mobilní aplikaci pro trénink hudebního sluchu. V teoretické části byla uvedena základní hudební teorie a bylo popsáno trénování hudebního sluchu. V následujících částech se práce věnovala analýze, ve které byly uvedeny funkční a nefunkční požadavky aplikace. Poté byl proveden rozbor konkurenčních aplikací.

Hlavním cílem mobilní aplikace bylo efektivně trénovat hudební sluch. To bylo dosaženo pomocí konfigurovatelného katalogu zvuků, podrobné statistiky a průvodcem. Implementace aplikace kladla důraz na rozšiřitelnost a pro tento účel byl zvolen návrhový vzor MVVM. Pro rozšiřitelnost v přidávání hudebních nástrojů byl zvolen návrhový vzor Visitor.

Prototypová mobilní aplikace by se dala vylepšit v mnoha směrech. Aplikace by se dala využít ve výuce na uměleckých školách. Dalším cílem bude zprovoznit projekt do reálného světa, což bude vyžadovat další změny, vylepšení a uživatelské testování. Jelikož se jedná o prototypovou aplikaci, která běží na lokálním zařízení, nejsou v práci implementované sekce ocenění a žebříček.

Možné rozšíření a vylepšení

Finální prototyp aplikace by se dal rozšířit a vylepšit v mnoha směrech. Jednak by se dala aplikace rozšířit svoji funkcionalitou, jednak také po grafické stránce.

Více Learn through play aktivit a propojení online

Hlavním motivačním aspektem je touha překonávat sebe sama a dohánět soupeře. Pro tento účel by bylo vhodné propojit aplikaci s účtem od Facebook nebo účtem od Google. Uživatel by mohl sledovat aktuální trendy svých vrstevníků, a tím se motivovat.

Velmi efektivním způsobem tréninku by pak bylo zabudování nějaké hry (Chcete být milionářem, dobyvatel.cz, Quiz planet, ale také Samsung Health apod.), přičemž by měl uživatel možnost vyzvat své přátele nebo jiné soupeře k získávání bodů, které by mohl využít k nákupu lepších zvuků a jiných vylepšení aplikace.

Hudební sluch a hudební cítění

Aplikace otevírá prostor pro další hudební disciplíny, které by zároveň trénovaly hudební cítění. Spočívalo by to například v rozpoznání rytmu nebo rozeznávání žánru skladby. Aplikace by mohla umožnit trénovat hudební sluch v celku, přičemž by uživatel měl za úkol provést dekompozici různých skladeb, kde by hádal například nástroje, které byly obsaženy ve zvukové stopě.

Obecná vylepšení

Aplikace by měla klást větší důraz na UI, UX a výkon. Aplikace by se dala přepsat nativním vývojem pro Android nebo iOS. To by přineslo hlavně optimalizaci ve výkonu. Dále vidím potenciál ve vývoji uživatelského rozhraní, do něhož by mohly být přidány animace nebo světlé a tmavé téma aj. Aplikace by mohla umožnit vícejazyčné balíčky, a tím by se aplikace rozšířila do světa.

Dalším problémem aplikace jsou nahrávky hudebních nástrojů, které nebyly kompletní a ve vysoké kvalitě. Některé tóny byly transformovány z referenčních tónů programem Audacity a mohly ztratit značnou kvalitu.

V neposlední řadě by se dal vylepšit samotný průvodce, který velice naivním způsobem generuje tréninky. Na tento problém by se dala využít umělá inteligence, neuronové sítě atd.

Seznam použitých zdrojů

1. EDWARDS, Scott. Musical Hearing: The Skill That Connects You With Music. In: *EarTrainingHQ* [online] [cit. 2019-04-04]. Dostupné z: <http://www.eartraininghq.com/musical-hearing-the-skill-that-connects-you-with-music>.
2. ZENKL, Luděk. *ABC: hudební nauky*. 2. vydání. Praha: Editio Bärenreiter, 2014. ISBN 978-80-86385-21-1.
3. PELÍŠEK, František. *Odborná konzultace: Hudební teorie a hudební sluch [osobní setkání]*. Klatovy: Základní umělecká škola Josefa Kličky v Klatovech. Datum konání: 2019-02-04.
4. ŠMIDÁK, Miron. *Absolutní sluch*. Praha, 2005. Dostupné také z: http://www.smidak.cz/smidak/doc/absolutni_sluch.pdf. Magisterská práce. Akademie múzických umění v Praze, Hudební fakulta, Katedra klávesových nástrojů. Vedoucí práce Emil Leichner.
5. FRANCISCO PAMPULHA. *The Ear Gym. Version 2.1.6* [software]. 2017 [cit. 2019-04-04]. Dostupné z: <https://play.google.com/store/apps/details?id=com.franciscop.francisco.eargym>.
6. EDUCKAPPSSV. *Perfect Ear. Version 3.8.6* [software]. 2013 [cit. 2019-04-04]. Dostupné z: <https://play.google.com/store/apps/details?id=com.evilduck.musiciankit>.
7. KAIZEN9 APPS. *Functional Ear Trainer. Version 2.0.7* [software]. 2017 [cit. 2019-04-04]. Dostupné z: <https://play.google.com/store/apps/details?id=com.kaizen9.fet.android>.
8. The Model-View-ViewModel Pattern. In: *Microsoft documentation* [online] [cit. 2019-04-04]. Dostupné z: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm>.

9. DAJBYCH, Václav. MVVM: Model-View-ViewModel. In: *DotnetPortal* [online] [cit. 2019-04-04]. Dostupné z: <https://www.dotnetportal.cz/clanek/4994/MVVM-Model-View-ViewModel>.
10. *The MVVM pattern* [online obrázek] [cit. 2019-04-04]. Dostupné z: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm-images/mvvm.png>.
11. *Xamarin* [online] [cit. 2019-04-04]. Dostupné z: <https://docs.microsoft.com/cs-cz/xamarin>.
12. *Visual Studio* [online] [cit. 2019-04-04]. Dostupné z: <https://visualstudio.microsoft.com>.
13. Xamarin Studio. In: *Documentation for Xamarin* [online] [cit. 2019-04-04]. Dostupné z: https://developer.xamarin.com/releases/studio/xamarin.studio_6.3/xamarin.studio_6.3.
14. *Audacity* [online] [cit. 2019-04-04]. Dostupné z: <https://www.audacityteam.org>.
15. An introduction to NuGet. In: *Microsoft documentation* [online] [cit. 2019-04-04]. Dostupné z: <https://docs.microsoft.com/en-us/nuget/what-is-nuget>.
16. *SQLite* [online] [cit. 2019-04-04]. Dostupné z: <https://www.sqlite.org>.
17. DENIEL, Aloïs. Microcharts. In: *GitHub* [online] [cit. 2019-04-04]. Dostupné z: <https://github.com/aloidsdeniel/Microcharts>.
18. NECIPOGLU, Enis. Xamarin.Forms.InputKit. In: *GitHub* [online] [cit. 2019-04-04]. Dostupné z: <https://github.com/enisn/Xamarin.Forms.InputKit>.
19. STEVENS, Adrian. Simple Audio Player. In: *GitHub* [online] [cit. 2019-04-04]. Dostupné z: <https://github.com/adrianstevens/Xamarin-Plugins/tree/master/SimpleAudioPlayer>.
20. ZOLIN, Simon. Audio Formats Comparison. In: *firmdev* [online] [cit. 2019-04-04]. Dostupné z: <http://fmedia.firmdev.com/audio-formats/>.
21. ONDERKA, Jan. *Pitch shifting audio signálů v reálném čase pomocí STFT na DSP procesoru*. Praha, 2018. Dostupné také z: <https://dspace.cvut.cz/bitstream/handle/10467/77279/F8-BP-2018-Onderka-Jan-thesis.pdf?sequence=-1&isAllowed=y>. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, Katedra číslicového návrhu. Vedoucí práce Martin Novotný.
22. *Musical Instrument Samples: Piano* [online] [cit. 2019-04-04]. Dostupné z: <http://theremin.music.uiowa.edu/MISpiano.html?fbclid=IwAR2gYSK6YjBN01VWIR6e6Dv3FMnSfc2GnuMdwZ3vHC9lsnZC66QRHbL0IMk>.

23. DO, Duc Huy. *Poskytnutí audio záznamů* [osobní setkání]. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií. Datum konání: 2019-03-15.

Seznam použitých zkratk

GUI Graphical User Interface

UI User Interface

UX User Experience

XAML Extensible Application Markup Language

Ukázky mobilní aplikace

The image displays two mobile application screens side-by-side. The left screen is titled 'Register' and features a dark background with white text and rounded input fields. It includes fields for Username, Password, Birthday (with the value '01/01/2000'), and Country (with a dropdown arrow). A 'REGISTER AND LOG IN' button is at the bottom. The right screen is titled 'Login' and also has a dark background. It includes fields for Username (with the value 'fitak') and Password (with masked characters '.....'). A 'Remember Me' checkbox is checked. Below the fields are 'LOG IN' and 'REGISTER' buttons. A 'Don't have account?' link is positioned above the 'REGISTER' button.

Register

Username:

Password:

Birthday:

01/01/2000

Country:

REGISTER AND LOG IN

Login

Username:

fitak

Password:

.....

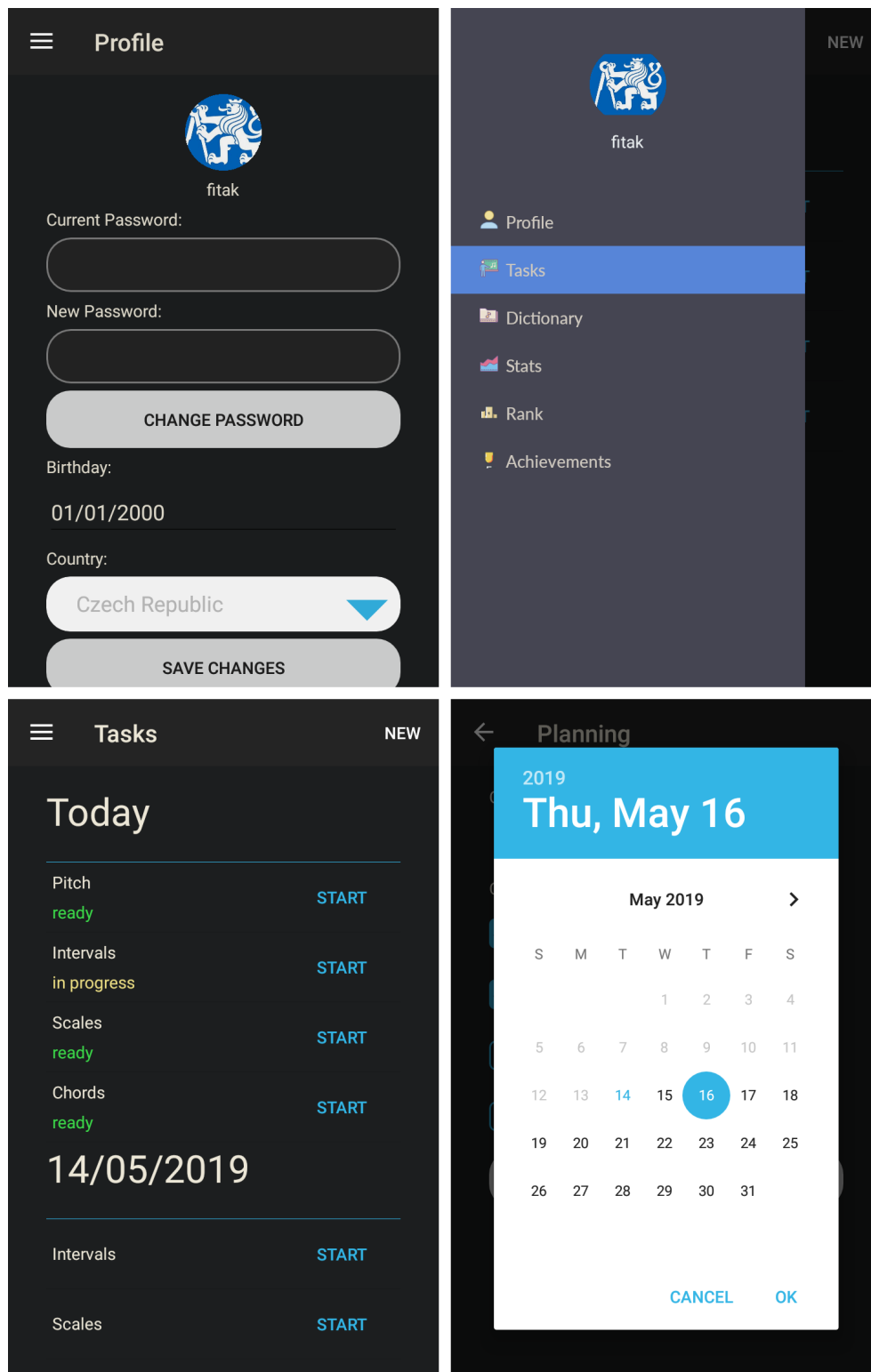
Remember Me

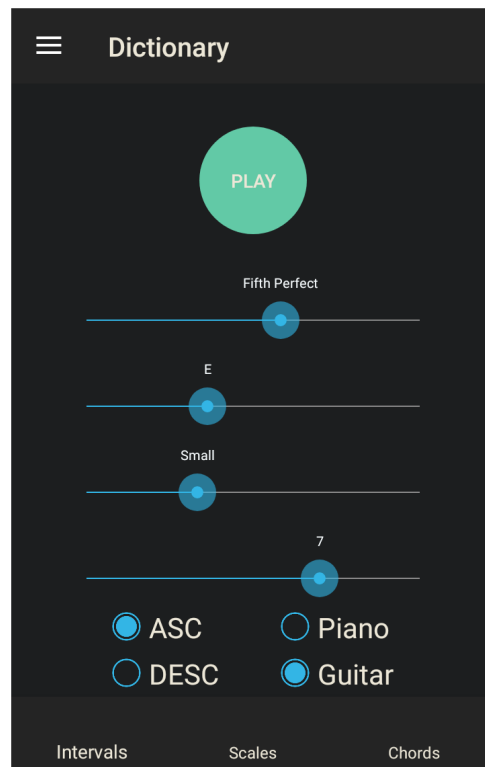
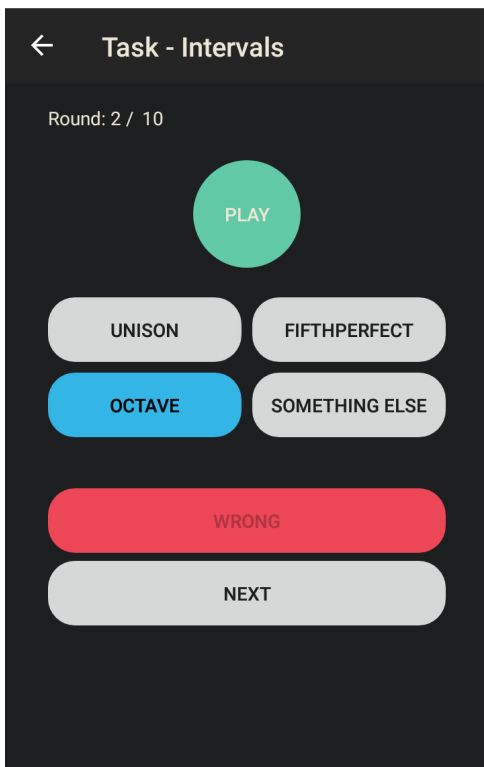
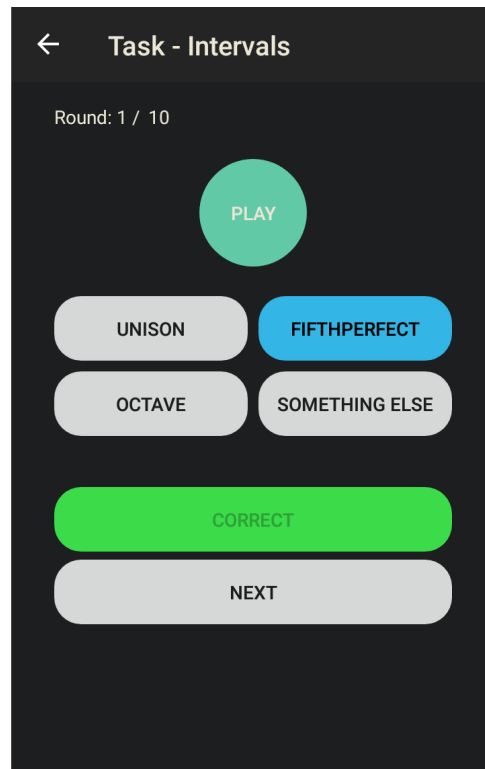
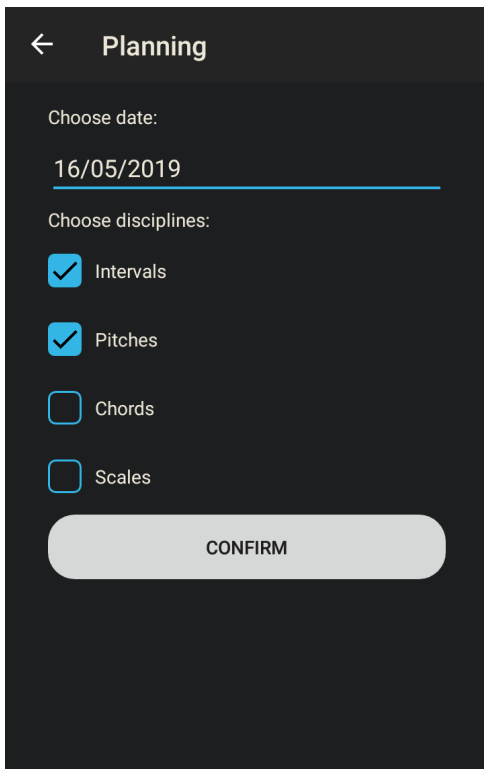
LOG IN

Don't have account?

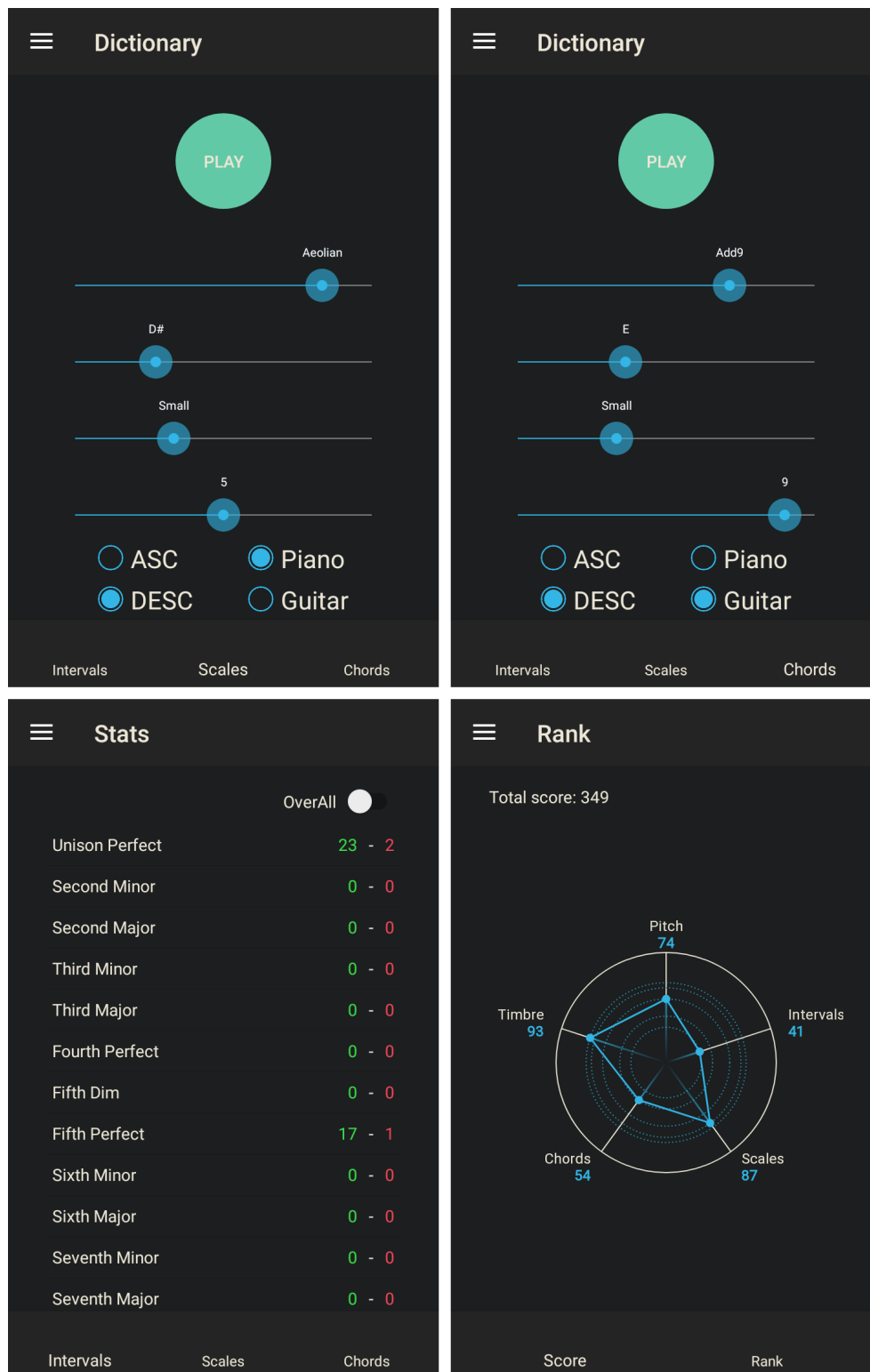
REGISTER

B. UKÁZKY MOBILNÍ APLIKACE





B. UKÁZKY MOBILNÍ APLIKACE



Obsah přiloženého USB disku

readme.txt.....	stručný popis obsahu USB disku
samples	adresář s ukázkami
├── images	adresář se snímky obrazovky aplikace
└── sample.mp4.....	video ukázka aplikace ve formátu MP4
src	
├── implementation.....	zdrojové kódy implementace
└── thesis.....	zdrojová forma práce ve formátu \LaTeX
text	text práce
└── thesis.pdf.....	text práce ve formátu PDF