



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**MOBILNÍ APLIKACE PRO SPRÁVU TVORBY  
HUDEBNÍ SKUPINY**

MOBILE APPLICATION FOR MANAGEMENT OF MUSIC BAND PRODUCTION

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**VEDOUCÍ PRÁCE**

SUPERVISOR

**MILOŠ DOLINSKÝ**

**Ing. JIŘÍ HYNEK**

BRNO 2017

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav informačních systémů

Akademický rok 2016/2017

**Zadání bakalářské práce**

Řešitel: **Dolinský Miloš**

Obor: Informační technologie

Téma: **Mobilní aplikace pro správu tvorby hudební skupiny**  
**Mobile Application for Management of Music Band Production**

Kategorie: Uživatelská rozhraní

**Pokyny:**

1. Provedte průzkum dostupných aplikací a služeb určených pro muzikanty a hudební skupiny. Analyzujte jejich výhody a nedostatky.
2. Seznamte se s principem tvorby mobilních aplikací pro operační systém Android.
3. Na základě pokynů vedoucího a poznatků získaných z bodů 1 a 2 navrhnete mobilní aplikaci určenou pro správu aktivit a nové tvorby hudební skupiny (snadná interpretace nových hudebních nápadů; nastavení nástrojů/aparátů/efektů; správa skladeb a koncertů). Spolupracujte s členy kapely Insane TV.
4. Navrženou aplikaci implementujte.
5. Otestujte přívětivost implementovaného uživatelského rozhraní.
6. Navrhnete možná rozšíření a nastiňte další vývoj projektu.

**Literatura:**

- Johnson, J.: *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Burlington: Morgan Kaufmann Publishers/Elsevier, 2010, ISBN 978-0-12-375030-3.
- Rogers, Y.; Sharp, H; Preece, J.: *Interaction Design: Beyond Human - Computer Interaction*. 3rd Edition. New York: Wiley, 2011, ISBN 978-0-470-66576-3.
- Introduction to Android. *Android Developers* [online]. [cit. 2015-10-23]. Dostupné z: <https://developer.android.com/guide/index.html>.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

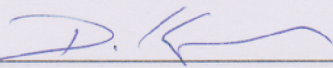
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Hynek Jiří, Ing.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav informačních systémů  
612 66 Brno, Božetěchova 2

  
doc. Dr. Ing. Dušan Kolář  
vedoucí ústavu

## Abstrakt

Tato práce popisuje celý proces vývoje aplikace MyBandSong, která slouží ke správě tvorby hudební skupiny. Součástí aplikace je také jednoduchý editor pro zápis hudby ve formě tabulatury. Popsány jsou jednotlivé etapy vývoje, od specifikace požadavků, až po návrh uživatelského rozhraní, jeho implementaci a konečné testování. Aplikace je implementována pro operační systém Android v programovacím jazyce Java. MyBandSong umožňuje skupinám jednoduše spravovat svou tvorbu a koncerty z jednoho místa, díky čemuž je možné ušetřit čas jak na kapelních zkouškách, tak při přípravě na koncerty.

## Abstract

This bachelor thesis describes entire process of development of application MyBandSong that serves to manage creations of a musical band. A part of application is also a simple editor for writing music in the form of a tablature. Individual phases of development are described – from the requirements specification to design of a user interface, its implementation and final testing. Application is implemented in Java language for Android operating system. With MyBandSong bands can easily manage their songs and concert from one place, which can save time on rehearsals or before concerts.

## Klíčová slova

Tabulatura, Android, aplikace, hudební skupina, zápis hudby, Java, návrh uživatelského rozhraní, správa tvorby

## Keywords

Tablature, Android, application, musical band, music notation, Java, user interface design, band manager

## Citace

DOLINSKÝ, Miloš. *Mobilní aplikace pro správu tvorby hudební skupiny*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Hynek Jiří.

# Mobilní aplikace pro správu tvorby hudební skupiny

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Hynka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Miloš Dolinský  
15. května 2017

## Poděkování

Tímto bych chtěl poděkovat Ing. Jiřímu Hynkovi za veškerou pomoc, které se mi od něj při tvorbě této bakalářské práce dostalo. Zároveň bych chtěl poděkovat své přítelkyni a rodičům, kteří mě při tvorbě práce podporovali. Poděkování si zaslouží i mí zvířecí přátelé, díky kterým bylo snadné překonat každou krizi.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Základy hudební teorie a terminologie</b>	<b>4</b>
2.1	Hudební notace . . . . .	4
2.2	Standardní hudební notace . . . . .	4
2.3	Tabulatura . . . . .	6
2.4	Technika hry . . . . .	8
2.5	Struktura písně . . . . .	8
<b>3</b>	<b>Specifikace požadavků a analýza existujících řešení</b>	<b>9</b>
3.1	Pozorování uživatelů . . . . .	9
3.2	Dotazník . . . . .	10
3.3	Specifikace požadavků . . . . .	11
3.4	Analýza existujících řešení . . . . .	12
3.4.1	Song Manager . . . . .	13
3.4.2	Songsterr . . . . .	13
3.4.3	Guitar Tabs X . . . . .	14
3.4.4	Tabify Guitar Composition . . . . .	14
3.4.5	TuxGuitar . . . . .	14
3.4.6	Lyric Pad . . . . .	14
3.4.7	Google kalendář . . . . .	15
3.4.8	Guitar Pro . . . . .	15
3.5	Vyhodnocení analýzy existujících řešení . . . . .	15
<b>4</b>	<b>Operační systém Android</b>	<b>17</b>
4.1	Základní informace . . . . .	18
4.2	Srovnání mobilních operačních systémů . . . . .	18
4.3	Architektura . . . . .	19
4.4	Android API . . . . .	20
4.5	Material Design . . . . .	21
4.6	Dostupné nástroje pro vývoj . . . . .	21
<b>5</b>	<b>Návrh aplikace</b>	<b>22</b>
5.1	Cílová skupina uživatelů . . . . .	22
5.2	Výběr názvu aplikace . . . . .	22
5.3	Konceptuální model . . . . .	23
5.3.1	Akce . . . . .	23
5.3.2	Matice objektů a akcí . . . . .	24

5.4	Uživatelské rozhraní . . . . .	25
5.4.1	Hlavní obrazovka . . . . .	25
5.4.2	Struktura písně . . . . .	26
5.4.3	Tabulatura . . . . .	26
5.4.4	Zápis hudby . . . . .	27
5.4.5	Zobrazení detailů položek . . . . .	28
5.4.6	Editační formuláře . . . . .	29
<b>6</b>	<b>Implementace</b>	<b>30</b>
6.1	Databáze . . . . .	30
6.2	Hlavní obrazovka a obrazovka struktury písně . . . . .	32
6.3	Seznamy hudebních položek . . . . .	33
6.4	Poznámky a text skladby . . . . .	33
6.5	Přidání a editace položky . . . . .	33
6.6	Tabulatura . . . . .	34
6.6.1	Model . . . . .	34
6.6.2	Zobrazení . . . . .	35
6.6.3	Ovládání . . . . .	37
<b>7</b>	<b>Testování</b>	<b>38</b>
7.1	Optimalizace . . . . .	38
7.2	Testování uživatelského rozhraní . . . . .	38
7.3	Vyhodnocení výsledků . . . . .	39
<b>8</b>	<b>Závěr</b>	<b>40</b>
	<b>Literatura</b>	<b>41</b>
	<b>Přílohy</b>	<b>43</b>
<b>A</b>	<b>Obsah CD</b>	<b>44</b>
<b>B</b>	<b>Snímky obrazovek aplikace</b>	<b>45</b>

# Kapitola 1

## Úvod

Na světě existuje spousta lidí, kteří hrají v nějaké hudební skupině. Každá tato skupina má určitý repertoár písní, což pro muzikanta znamená spoustu informací k zapamatování. Pro udržení všech těchto informací v mysli se členové kapely jednou za čas sejdou a spolu si zahrají všechny své písně, popřípadě pracují na nových. Často se stane, že na těchto zkouškách spontánně vzniknou nové nápady. Například baskytarista obmění svoji linku v refrénu, bubeník přidá na konec sloky nový přechod, vznikají nové části písniček a nevyhovující se odstraňují.

Člověk se denně vyrovnává s ohromným množstvím informací o různé informativní hodnotě, a tak se snadno stane, že nové nápady zapomene. Papíry s poznámkami nebo texty písniček se snadno ztrácejí a také nejsou zrovna ekologicky přívětivým řešením. Tím pádem je spousta zajímavých nápadů do další zkoušky nadobro ztracena. Tento problém v některých případech umocňuje i frekvence zkoušek. Některé kapely je mají častěji, jiné zřídka.

Téměř každý člověk v dnešní době vlastní chytrý telefon (v roce 2016 je to přes 2 miliardy uživatelů<sup>1</sup>), který má skoro vždy u sebe. Proč tedy nevyužít tohoto potenciálu a nespravovat svou hudební tvorbu z jednoho přístroje?

Mým cílem je navrhnout a implementovat aplikaci, která hudebním skupinám usnadní správu nad její vlastní tvorbu. Aplikaci, kde jsou všechny důležité informace o písni dostupné z jednoho místa, což usnadní práci na zkouškách, pomůže rychle osvěžit paměť před koncerty, kdy je tréma v plných obrátkách, nebo pomůže s učením současného repertoáru novým členům kapely. A protože jak já samotný, tak i ostatní členové naší kapely mají chytrý telefon s operačním systémem Android, je tato aplikace implementována právě pro něj.

Kapitola 2 čtenáře seznámí se základy hudební teorie a terminologie nutné pro další orientaci v textu této bakalářské práce. V kapitole 3 jsou rozepsány poznatky z analýzy trhu a existujících řešení. Dále je pak sepsána specifikace požadavků na aplikaci. Kapitola 4 je zaměřena na seznámení s mobilním operačním systémem Android. Kapitola 5 se zabývá již samotným návrhem uživatelského rozhraní a tomu, jak bylo při návrhu postupováno. Implementace aplikace je popsána v kapitole 6, testování pak v kapitole 7. Kapitola 8 je zároveň kapitolou poslední a obsahuje závěr.

---

<sup>1</sup>Viz <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>.

## Kapitola 2

# Základy hudební teorie a terminologie

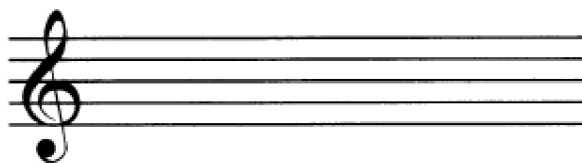
Tato kapitola se zabývá základy hudební teorie, která je nezbytná pro pochopení dalšího textu této bakalářské práce. Cílem kapitoly je vymezení nejdůležitějších hudebních pojmů a probrání hlavních pilířů zápisu hudby. Informace zde popsané vycházejí z knih Essential dictionary of music notation [8] a Historical dictionary of baroque music [11].

### 2.1 Hudební notace

*Hudební notace* je jazyk, a jako k takovému se k němu musí i přistupovat. Čím víc se notace blíží všeobecně známým pravidlům, tím snadněji a úspěšněji bude hudba reprodukována [8]. Hudební notace je tedy způsob jak zapsat hudbu.

### 2.2 Standardní hudební notace

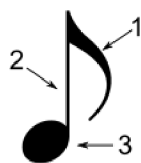
Nejrozšířenějším typem hudebního zápisu je *standardní notace* – *notová osnova ve spojení s notami*. Notovou osnovu (vyobrazena na obrázku 2.1) představuje pět vodorovných linek poskládaných rovnoběžně pod sebou.



Obrázek 2.1: Notová osnova.

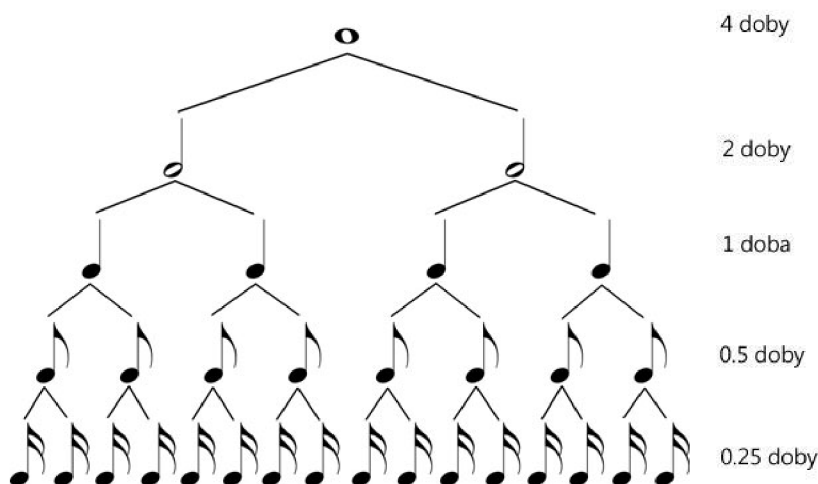
Do notové osnovy se vpisují *noty*, které jsou základním stavebním kamenem hudby. Složení noty znázorňuje obrázek 2.2. *Bříško* noty (na obrázku označené číslem 3) je přibližně o velikosti mezery mezi dvěma řádky notové osnovy. Z bříška vede *nožička* (na obrázku označené číslem 2). Z nožičky může ještě volitelně vést *praporek* (na obrázku znázorněné číslem 1). Složení noty určuje délku tónu, neboli dobu, po kterou bude tón znít.





Obrázek 2.2: Složení noty. 1 – praporek, 2 – nožička, 3 – bříško <sup>1</sup>.

Noty se podle tvaru dělí na několik druhů. Nota *celá* zní po čtyři doby (doba než napočítáme do čtyř). Následuje nota *půlová*, počítaná na dvě doby. Další notu opět získáme rozpůlením doby noty předchozí (viz obrázek 2.3) – nota *čtvrtová* na jednu dobu, *osminová* na půl doby, *šestnáctinová* na čtvrt doby, a tak dále. Délka noty se dá prodloužit o polovinu doby symbolem tečky umístěným těsně vedle noty.



Obrázek 2.3: Délky not a jejich dělení. Na vrcholu je celá nota. Následují noty půlové, čtvrtové, osminové a šestnáctinové <sup>3</sup>.

Dalším důležitým aspektem hudby jsou části, kde se nic nehraje. Tyto úseky se vyznačují *pomlkami* (vyobrazeny na obrázku 2.4) a jsou rozděleny úplně stejně jako délky not – celé, půlové, čtvrtové, a tak dále.

Délku noty nebo pomlky také ovlivňuje *tempo*, které je určováno v jednotkách úderů za minutu, neboli *bpm* (*beats per minute*). Noty a pomlky se shromažďují do *taktů*, což jsou krátké rytmické úseky skladby. Takt je určen počtem dob a délkou základní doby. Značí se jako dvě čísla pod sebou na začátku notové osnovy (viz obrázek 2.5) a je známé pod pojmem *metrum*. Například 7/8 znamená, že jeden takt je dlouhý sedm osminových dob. Jednotlivé takty se v notové osnově oddělují kolmou čarou.

<sup>1</sup>Dostupné z <https://cs.wikipedia.org/wiki/Nota>.

<sup>3</sup>Dostupné z <http://mtschoools.net/noteandrestvalues?i=1>.



Obrázek 2.4: Pomlky seřazené sestupně podle své délky. Nejvíc vlevo je celá, pak půlová, čtvrtová, atd.



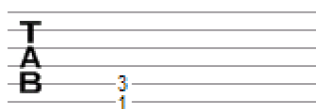
Obrázek 2.5: Označení taktu, neboli *metrum*.

Pro lepší vizuální orientaci v notaci se skupiny krátkých not spojují *trámci*. Mezi krátké noty patří noty menší než čtvrtové. Nejčastěji se spojují skupiny not tak, aby jejich celková délka dala délku základní doby taktu. Na obrázku 2.5 je vyobrazení spojení not v čtyřčtvrtovém taktu.

Vertikální umístění noty v notové osnově určuje výšku hraného tónu – tedy čím výše se nota v notové osnově nachází, tím vyšší je tón. Noty se píší buď do mezer mezi linkami, nebo přímo na linky tak, že linka notu protkává jejím středem. Dále se noty mohou psát i pod nebo nad notovou osnovu, k čemuž se používají takzvané *pomocné linky*. Pro reprodukování takto zaznamenané hudby je nutná znalost toho, jaké tóny symbolizují jednotlivé pozice not v notové osnově. Existuje ještě spousta dalších symbolů pro zápis hudby ve standardní notaci, avšak jejich znalost pro orientaci v textu bakalářské práce není potřebná.

## 2.3 Tabulatura

*Tabulatura* je oproti notové osnově daleko jednodušší. Pro její pochopení a orientaci v ní není potřeba takřka žádných znalostí hudební teorie. Od standardní notace se liší tím, že neukazuje, které tóny hrát, ale kam dát na nástroji prsty. Jedná se tedy o druh hudební notace navržené pro specifické nástroje (nejčastěji strunné a dechové) [11]. Stejně jako v případě standardní notace, v tabulatuře existují linky, na které se značky píší, ovšem s tím rozdílem, že jejich počet je dán možnostmi daného nástroje. Například pro šestistrunnou kytaru bude mít tabulatura šest linek, kde každá z těchto linek symbolizuje jednu strunu na kytarě. Pro flétnu by to byla jedna linka. Na tyto linky se poté umísťují čísla podle toho, kolikátý otvor na dechovém nástroji nebo pražec na strunném nástroji se má stisknout. Takže pro zaznamenání hudby, která vznikne stisknutím prvního pražce na první struně a současně třetího pražce na druhé struně a následným zahráním, se do tabulatury na první linku odspodu napíše číslo jedna a na druhou linku odspodu v témže sloupci číslo tři (viz obrázek 2.6). Z toho vyplývá, že pro čtení tabulatury není nutná znalost not a jejich pozic v notové osnově. Malý rozdíl ve značení je v tabulatuře pro bicí nástroje, kde jsou jednotlivé komponenty bicích namísto čísel označeny písmeny. Například *sn* pro rytmický buben (snare drum), *b* pro basový buben a podobně.



Obrázek 2.6: Zápis hudby pomocí tabulatury.

Čistá tabulatura ovšem neposkytuje takové možnosti zápisu hudby jako notová osnova s notami. Tabulatura pouze ukazuje jak docílit daného tónu, ne jak dlouho má znít. Proto se často nad jednotlivé sloupce čísel zaznamenávají ještě rytmické značky. Ty nejsou nic jiného než noty, avšak s tím rozdílem, že v tabulatuře se nebere vůbec v potaz jejich umístění, ale pouze jejich tvar, který říká, jak dlouho bude jaký tón znít. Pro vyznačení tichého místa se v rytmické části tabulatury používají pomlky. Jak je možné vidět na obrázku 2.7, přímo v tabulatuře se nic nevyznačuje. Stejně jako ve standardní notaci, i zde se pro zpřehlednění dělí úseky skladby do taktů. Tabulatura také dokáže zaznačit i různé techniky hraní, toto značení ale není nijak standardizováno a napříč tabulaturami se různě liší.



Obrázek 2.7: Tabulatura společně s rytmickou sekci.

Mohlo by se zdát, že tabulatura je výtvarným produktem dnešní doby, avšak nejstarší známá tabulatura pochází ze 14. století, známá pod názvem Robertsbridge Codex<sup>4</sup>. V době baroka se tabulatura stala oblíbeným způsobem zaznamenání hudby pro kytary, loutny a další podobné strunné nástroje [11].



Obrázek 2.8: Všechny zmíněné pojmy použité pospolu – tečka, noty, označení taktu.

<sup>4</sup>Hudební rukopis ze 14. století pro klavír

## 2.4 Technika hry

Je spousta možností jak ovlivnit zvuk zahraného tónu. Pro jeho ozvláštňení se využívá různých technik hry. A protože techniky hry mohou radikálně změnit konečný zvuk a dojem ze zahraného tónu, je potřeba je do hudební notace zaznamenat. Různých technik hraní na odlišné hudební nástroje je nespočet, proto následuje popis jen těch nejzákladnějších. Popis technik je inspirován knihou Technika hry na elektrickou kytaru [13], nicméně techniky jsou aplikovatelné na širší spektrum nástrojů. Mezi nejčastější patří:

- *vibrato*, což je technika spočívající v mírném plynulém opětovném zvyšování a snižování tónu, čímž se dosáhne jeho rozkmitání,
- *bending*, jenž je název pro plynulé a postupné zvýšení tónu vytáhnutím struny,
- *palm muting*, kde tón vzniká tlumením dozvuku strun zápěstím pravé ruky, čímž se dosáhne charakteristického zvuku škrabotu trsátka o závin strun a tlumeného basového dozvuku,
- *hammer on*, kde tón vzniká přírazem prstu na pražec,
- *pull-off*, kde tón vzniká držením pražce a následným rychlým odtrhem prstu,
- *legato*, což je souvislá a pravidelná kombinace přírazu (*hammer on*) a odtrhu (*pull-off*),
- *slide*, kde tón vzniká skluzem, nebo dojetím prstu nad cílový pražec

## 2.5 Struktura písně

Každá píseň se dá zpravidla rozložit na jednotlivé části s určitým rysem, které se, pouze v malých obměnách, mohou opakovat napříč celou písní. Tyto části určují strukturu písně a pouhým pohledem na ně lze říct, kde píseň graduje, kde je nějaký stále se opakující motiv a podobně. Existuje několik ustálených pojmenování pro určité části, jako například intro, sloka, refrén. Při nahrávání ve studiu se pro lepší přehlednost nad nahrávanou písní často používá tabulka, kde jsou popisy řádků pojmenované po jednotlivých hudebních nástrojích a popisy sloupců po chronologicky seřazených částech písně. Poté se do jednotlivých buněk tabulky pro konkrétní části písně píše nastavení nástroje a aparátu, popřípadě použité efektové krabičky a podobně. Příkladem může být tabulka 2.1.

	Intro	Sloka	Refrén
<b>Kytara</b>	-	Hlasitost na úrovni 5	Čistý kanál, hlasitost na úrovni 6
<b>Basa</b>	-	Kompresor	Kompresor
<b>Hlavní zpěv</b>	Přidat reverb	Lehce zkreslit	-

Tabulka 2.1: Stručný a ilustrativní příklad tabulky sekcí písně a použitých nástrojů. Reálně by tabulka obsahovala daleko více sekcí i nástrojů a poznámky uvnitř tabulky by mohly být více detailní.

## Kapitola 3

# Specifikace požadavků a analýza existujících řešení

V následující kapitole nejprve specifikuji požadavky na aplikaci pro správu tvorby hudební skupiny. Na základě této specifikace poté provedu analýzu existujících řešení, kde se zaměřím hlavně na splnění určených požadavků. Tyto požadavky jsou sepsány speciálně pro potřeby skupiny InsaneTV na základě pozorování a komunikace s jejími členy (mezi které také patřím). Nicméně podobné požadavky může mít více hudebních skupin.

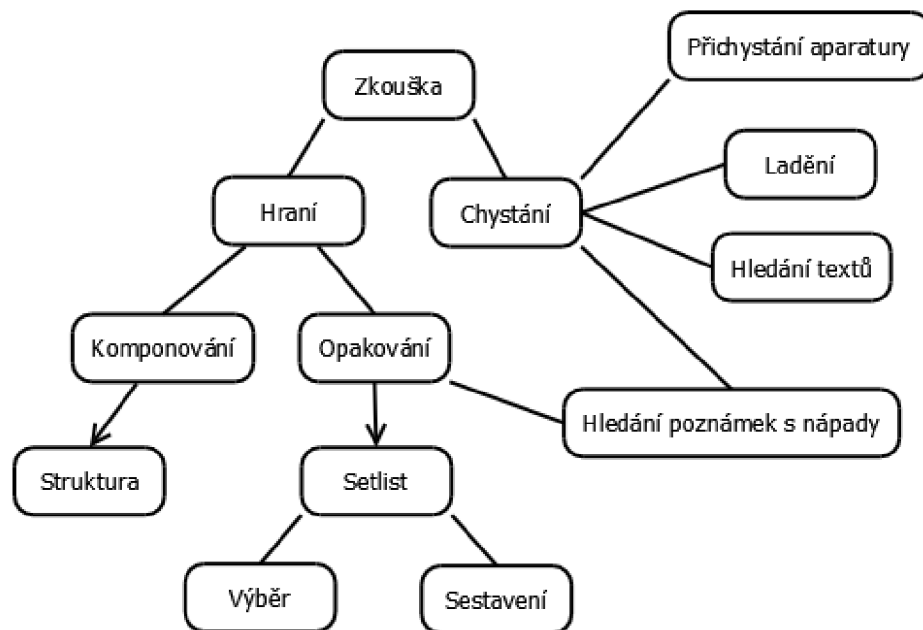
### 3.1 Pozorování uživatelů

Za téměř tři roky mého působení ve skupině InsaneTV jsme společně odehráli desítky koncertů a strávili stovky hodin ve zkušebně trénování a komponování nových skladeb. Díky tomu jsem měl možnost, ať už jako přímý pozorovatel, tak i jako účastník detailně nahlédnout do činností a problémů kapely.

Na zkouškách vznikla spousta hudebních nápadů, které kolikrát přišla vniveč jen proto, že se nikam nezaznamenala. Tento problém členové skupiny částečně vyřešili tím, že si nápady ve formě tabulatury zaznamenali na papír. Ovšem značné procento těchto poznámek neobsahovalo informace o délkách tónů, takže u složitějších písní nebylo možné odvodit rytmiku. Další nevýhodou tohoto řešení je spousta papírů k uchování, ve kterých se obtížně vyhledává. Když se k nim ještě přidají papíry s texty písní, zabere několik minut než se nalezne hledaná poznámka. Chystání na hraní a samotné hraní shrnuje diagram na obrázku 6.10.

Příprava na koncerty často předcházela činnost vyhledávání základních informací o nich z e-mailových konverzací (místo a čas koncertu, délka vystoupení). Samotné chystání se poté skládalo mimo jiné z těchto činností:

- sestavování seznamu skladeb o potřebné délce,
- sepsání seznamu skladeb na papír,
- hledání textů ke hraným skladbám,
- přepisování nečitelných nebo ztracených textů



Obrázek 3.1: Zkouška se dělí na dvě činnosti – chystání věcí potřebných pro hraní a samotné hraní. Kapela si po příchodu do zkušebny nejprve nachystá vybavení a vyhledá či setřídí potřebné poznámky (písně, texty). Hlavní činností zkoušky je hraní, které se dělí na komponování nových písní (případně nahrávání) nebo opakování těch již hotových a naučených. Skupina opakuje skladby na základě seznamů skladeb – buď zvolí již připravený nebo sestaví nový. K opakování starších písniček skupina využívá poznámek se zaznamenanými hudebními nápady, podle kterých si hudbu připomenou. Pro rychlejší zorientování v nově komponované písni skupina využívá papíru se sepsanou strukturou písně. Do jednotlivých sekcí struktury také zaznamenávají poznámky s použitým nastavením aparátů a efekto- vých krabiček, které v průběhu dalších zkoušek různě přepisují novými poznatkami.

Na samotných koncertech v některých případech nastupovala tréma a s ní spojené negativní následky. Nebylo tedy výjimkou, že si někteří členové nemohli vzpomenout ani na pečlivě nacvičené pasáže. Před vystoupením si členové skupiny nastavovali zvuk na aparátích a efekty na svých efekto- vých pedálech. Nastavení aparátu často bývalo narozdíl od efekto- vých pedálů stále po celou dobu koncertu. Každá píseň v mnoha případech využívá různých nastavení efektů, takže se výjimečně stávalo, že se na nějaký efekt opomnělo.

Aparát a efekty bylo také potřeba správně nastavit před nahráváním skladeb. Aby se při pokračování nahrávací seance nemuselo správné nastavení hledat znovu, používala skupina tabulku podobnou tabulce 2.1. Do této tabulky byly zapisovány nejen informace o nastavení, ale také různé poznámky, postřehy a návrhy na zlepšení.

## 3.2 Dotazník

Dalším krokem průzkumu bylo zjistit technickou úroveň budoucích uživatelů. Pro tyto účely jsem sestavil krátký dotazník, který obsahoval tyto otázky:

- Máte zkušenosti s libovolným editorem tabulatur na jakémkoliv platformě? Jakým?

- Jak dlouho již vlastníte chytrý telefon?
- Používáte ve svém chytrém telefonu aplikace? Na jaké denní bázi?

Respondentů bylo dvacet, z toho všichni jsou aktivními členy nějakého hudebního uskupení. Všichni respondenti vlastní chytrý telefon minimálně dva roky z čehož lze usoudit, že již mají alespoň částečně v podvědomí gesta, kterými se chytrý telefon ovládá. Většina tázaných na třetí otázku odpověděla, že téměř na denní bázi využívá různých aplikací – nejčastěji hudební přehrávač. Všichni respondenti mají také zkušenosti s editorem tabulatur Guitar Pro 5, který je dostupný pro počítače. Tomuto programu se budu ještě věnovat v analýze existujících řešení.

### 3.3 Specifikace požadavků

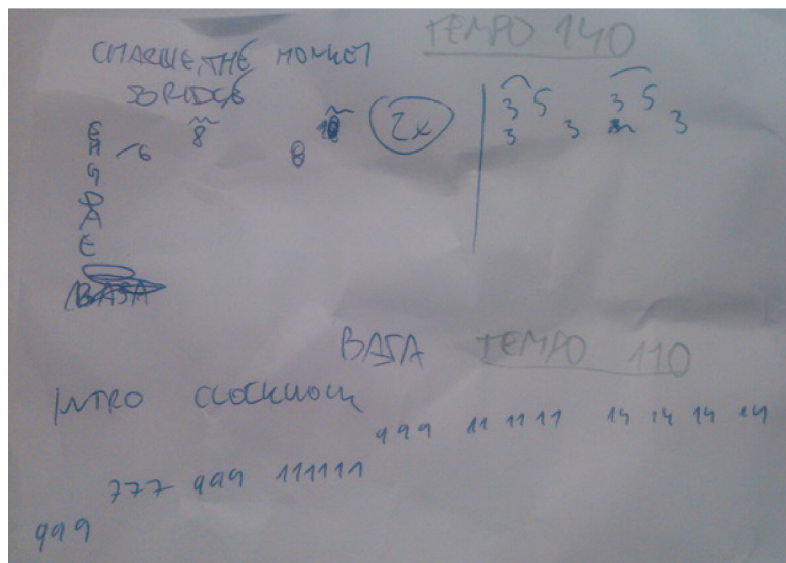
Na výslednou aplikaci jsem na základě analýzy poznatků z pozorování navrhl tyto požadavky:

1. vytváření seznamů skladeb,
2. přehled základních informací o písni,
3. zobrazení celkové délky seznamu skladeb v minutách,
4. přehled základních informací o koncertech,
5. editace a čtení textů k písním,
6. jednoduché zaznamenání hudebních nápadů i celých písní ve formě tabulatury společně s rytmickou sekcí,
7. náhled na celou strukturu písně s možností zaznamenání poznámek k jednotlivým sekcím struktury skladby

Hlavním smyslem všech požadavků je snížit režii činností spojených se správou tvorby kapely. Z přímého pozorování kapely totiž vychází najevo, že největším problémem je organizace poznámek a s ní spojená velká spotřeba papíru. Příklad poznámky vytvořené kapelou je na obrázku 3.2. Dostupnost všech potřebných informací z jednoho místa by činnosti skupiny značně usnadnila a urychlila. Jako řešení problematice organizace poznámek se nabízí jejich uspořádání do seznamů, což je podstatou požadavku 1. Tyto seznamy mohou sloužit i jako podklady ke koncertům nebo jako vodítko k tomu, co se bude cvičit další zkoušku. Ke každé uložené písni by měl být dostupný přehled základních informací, jako je název písně, název interpreta, tempo nebo délka. Proto je navrhnut požadavek 2.

Požadavky 3 a 4 skupině usnadní chystání na koncerty. Uvedení celkové délky seznamu skladeb umožní jednoduše sestavit seznam písní o potřebné délce. Požadavek 4 zamezí situacím, kdy je kapela nucena vyhledávat informace ze starých konverzací, případně odjinud. Mezi informace, které je o koncertu nutné vědět patří název, datum, čas a místo konání a seznam skladeb, které se budou hrát. Požadavek 5 kromě chystání na koncerty usnadní a zpřehlední také přepisování textů.

Splnění požadavku 6 bude nejkritičtější. Zapisování hudebních nápadů společně se všemi symboly musí probíhat intuitivně a rychle, aby to příliš nezdržovalo od hraní na nástroj. Výsledná tabulatura musí být přehledná a zároveň zobrazovat všechny důležité informace



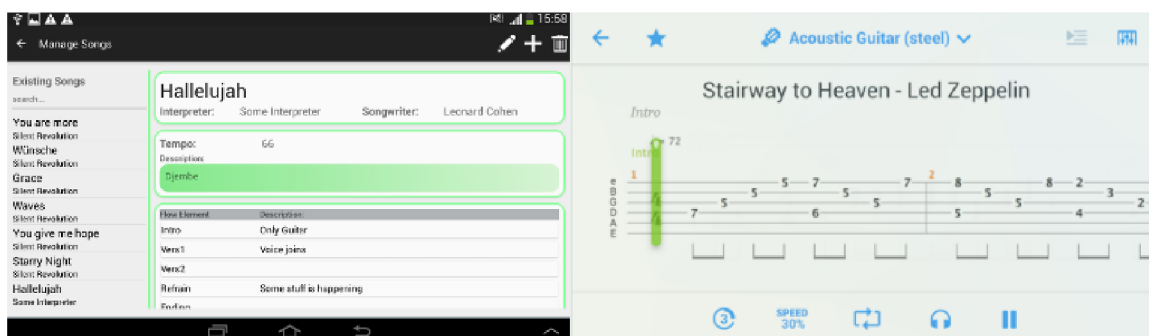
Obrázek 3.2: Ukázka poznámek vytvořených kapelou na zkoušce.

včetně rytmické sekce. Zápis do tabulatury musí poskytnout takové možnosti, aby muzikantům po zpětném nahlédnutí nedělalo problém hudbu interpretovat.

Požadavek 7 umožní muzikantovi pohledem zjistit jak je píseň zaranžovaná, kde je potenciální gradace, apod. Z toho vyplývá, že struktura písně musí být vyobrazena přehledným způsobem. Prostor pro poznámky u každé sekce musí být dostatečně velký, aby bylo možné uložit všechny potřebné informace.

### 3.4 Analýza existujících řešení

Analýza podobných existujících řešení je zaměřena především na účel každé aplikace, přívětivost uživatelského rozhraní, ale také její dostupnost. Důraz je kladen především na splnění specifikovaných požadavků. U každé aplikace jsou také vyhodnoceny její klady a zápory.



Obrázek 3.3: Analyzované aplikace. Vlevo Song Manager, vpravo Songsterr <sup>1</sup>.



### 3.4.1 Song Manager

Aplikace slouží ke správě písní. Ke každé skladbě lze zadat informace jako název písně a interpreta, tempo a popis. Aplikace umožňuje i náhled na strukturu písně a zaznamenávání poznámek k jednotlivým sekcím. Song Manager (na obrázku 6.12 vlevo) je dostupná zdarma a neobsahuje reklamy.

Mezi výhody patří přehledné a jednoduché grafické provedení. V aplikaci se snadno orientuje a snadno se ovládá. Písně lze také přidávat do seznamů skladeb, což usnadňuje orientaci. Velkou výhodou je zmiňovaný náhled na strukturu písně.

Mezi nevýhody naopak patří, že poznámky k sekcím skladby mohou být pouze stručné (o rozsahu přibližně jedné věty). Na nastavení nástrojů a podobné delší záznamy není prostor. Velkým nedostatek je nekompatibilita se staršími verzemi Android. U seznamů skladeb také není informace o celkové době trvání. Také chybí možnost zaznamenávat vlastní hudební nápady ve formě jakékoliv notace. Stejně tak tomu je u textu písně.

### 3.4.2 Songsterr

Songsterr (na obrázku 6.12 vpravo) je aplikace sloužící k učení písní prostřednictvím tabulatur. Za poplatek 148,99 Kč má uživatel k dispozici celou aplikaci s doživotním přístupem ke všem tabulaturám.

Mezi přednosti aplikace patří velmi kvalitní zpracování a intuitivní ovládání. Ke každé skladbě jsou uloženy základní informace jako název a tempo. Nechybí ani možnost zobrazení textu skladby. Poskytuje i možnost přehrání tabulatury. Tabulaturu lze zobrazit pro různé nástroje. Vyobrazeny jsou také informace o délkách tónů. Jednotlivé sekce písně jsou vyznačeny přímo nad tabulaturou.

Mezi nevýhody patří nemožnost seskupování písní do seznamů skladeb. Písně lze označit pouze jako oblíbené. Aplikace slouží jako výukový software, nelze tedy zaznamenávat vlastní hudbu. Aplikace také nespĺňuje požadavek 7.



Obrázek 3.4: Analyzované aplikace. Zleva TuxGuitar, Tabify Guitar Composition, Guitar Tabs X, Lyric Pad <sup>3</sup>.

<sup>1</sup>Dostupné z <https://play.google.com/store>.

### 3.4.3 Guitar Tabs X

Aplikace slouží k zaznamenání skladeb ve formě tabulatury. Je dostupná zdarma bez reklam. Výhodou je, že nechybí informace o délce tónu symbolizovaného tvarem noty. Další je dostupná informace o názvu písně a tempu. V aplikaci se jednoduše orientuje, grafický styl je jednotný a přehledný. Zápis tabulatury v této aplikaci je vidět na obrázku 3.4.

Mezi nevýhody patří omezené možnosti tabulatury – skladby lze zaznamenávat pouze pro kytaru nebo baskytaru (liší se počtem strun). Samotné psaní tabulatury je neintuitivní a složité. Pro psaní má aplikace vlastní klávesnici, která zabírá téměř půl displeje. Občas se objevuje až na druhé či třetí zavolání. Při každém dotyku obrazovky se navíc naspodu displeje zobrazuje implicitní ovládací klávesy systému Android, takže prvky, které jsou v aplikaci na tom místě, nejsou dostupné. Skladby nelze seskupovat do seznamů. Také není možné zobrazit strukturu skladby či text písně.

### 3.4.4 Tabify Guitar Composition

Slouží k zaznamenávání a prohlížení hudebních nápadů ve formě tabulatury. Tabulatury lze ukládat jako písně a poté jimi procházet. Aplikace je dostupná zdarma avšak s jistými omezeními (limit uložených písní). Obsahuje také reklamy, které zmizí po nákupu aplikace. Nespornou výhodou je intuitivní a jednoduché vytváření a editace tabulatur. Aplikace se ovládá snadno a přehledně. Zápis tabulatury v této aplikaci je vidět na obrázku 3.4.

Jak už název napovídá, tabulatury lze vytvářet pouze pro kytaru, což je první nevýhodou aplikace. Dalším nedostatkem je, že ke každé písni lze vytvořit pouze jednu stopu. Pokud by uživatel chtěl více kytar u jedné písni, musí vytvořit více písní. Chybí také informace o délkách tónů. Aplikace poskytuje funkci zaznamenávání hudby pomocí mikrofону a následné převedení zaznamenaných tónů do formy tabulatury. Konverze je však velice nedokonalá a pomalá. Ve výsledku tedy tato funkce působí zbytečně. Mezi další nedostatky patří absence přehledu o struktuře písně a nemožnost seskupování skladeb do seznamů.

### 3.4.5 TuxGuitar

Tux Guitar (na obrázku 3.4 vlevo) je jednoduchý editor tabulatur určený pro zaznamenávání hudby pro strunné nástroje. Aplikace obsahuje reklamy a je dostupná bezplatně.

Mezi přednostmi aplikace patří minimalistické a přehledné uživatelské rozhraní. Aplikace pracuje se svou vlastní klávesnicí, díky které je zadávání čísel pražců do tabulatury intuitivní. Nad řádkem tabulatury je ještě vyobrazena standardní hudební notace. Nechybí tedy informace o délkách tónů.

Pokud ovšem uživatel chce změnit nastavní skladby (změna taktu, přidávání taktů, přidávání zvukových efektů a technik) musí vytoženou volbu hledat v dlouhých seznamech různých technik, nastavení, efektů, apod. To velice zpomaluje práci s aplikací. Aplikace také nesplňuje požadavek 6, protože tabulatury lze vytvářet pouze pro strunné nástroje. Opět chybí informace o struktuře písně a textu skladby, takže není splněn ani požadavek 7 a 5. Skladby nelze seskupovat do seznamů.

### 3.4.6 Lyric Pad

Aplikace slouží ke správě textů k písním. Je dostupná ve dvou verzích, a to placené a bezplatné. Liší se poskytnutou funkcionalitou. V bezplatné verzi je omezená kapacita uložených

<sup>3</sup>Dostupné z <https://play.google.com/store>.

skladeb, nelze vytvářet seznamy a na displeji je přítomná zpráva informující, že se jedná o zkušební verzi. Reklamy však aplikace neobsahuje. Placená verze stojí 149,99 Kč.

K jednotlivým skladbám lze zadávat rozsáhlé informace jak název písně, název interpreta, žánr, délku a tempo. Výhodou je možnost nastavení automatického posouvání textu skladby. Editace textu je jednoduchá a funguje jako klasický poznámkový blok (viz obrázek 3.4). Další výhodou je možnost řazení písní do seznamů (ovšem pouze v placené verzi).

Hlavní nabídka a ovládání aplikace působí překombinovaným dojmem z důvodu přítomnosti spousty různých tlačítek, které často vedou na jednu stejnou obrazovku. Aplikace také nesplňuje většinu požadavků, protože slouží pouze ke správě textů písní.

### 3.4.7 Google kalendář

Žádná z výše uvedených aplikací neumožňuje jednoduchou správu koncertů či událostí. Pro tyto potřeby dokáže posloužit Google kalendář, který se dá spárovat i s mobilním zařízením. Na nadcházející naplánované události upozorní ve stavové liště telefonu. Vytvořená událost se dá sdílet mezi vybranou skupinou lidí (v tomto případě mezi členy kapely). K události lze napsat popis, přidat místo, čas a nahrát libovolnou přílohu. Uživatelské rozhraní je velice intuitivní a přehledné. Nicméně aplikace není nijak zaměřena na správu nebo zapisování skladeb, tudíž nesplňuje většinu požadavků.

### 3.4.8 Guitar Pro

Guitar Pro je sice počítačovým programem, nicméně je mezi muzikanty velice rozšířený a oblíbený. Využívaný je jak amatéry, tak i slavnými muzikanty a počet písní, které jsou v něm zaznamenány je čítán na miliony. Software slouží k zapisování a přehrávání tabulatur.

Psaní tabulatur je v tomto programu velice jednoduché a intuitivní. Lze zaznamenat tabulaturu společně s rytmikou pro libovolný nástroj včetně technik hraní. Zapsanou hudbu lze i přehrát. K písním lze uložit i jejich texty. K dispozici je také náhled na strukturu.

Program však neslouží k organizaci písní a koncertů. Také nelze zapisovat poznámky k sekcím struktury a vytvářet seznamy skladeb.

## 3.5 Vyhodnocení analýzy existujících řešení

Z analýzy vyplývá, že na trhu neexistuje aplikace, která by splňovala všechny výše specifikované požadavky. Existuje velké množství aplikací zaměřených spíše na implementaci konkrétního bodu ze specifikovaných požadavků.

Z hlediska zobrazení písní ve formě tabulatury se specifikovaným požadavkům nejvíce přiblížila aplikace Songsterr. Jak je vidět na obrázku 6.12, vyobrazení tabulatury je provedeno velmi kvalitně a přehledně. Všechny potřebné ovládací prvky jsou dostupné z jedné obrazovky. Zajímavě řešené je zobrazení sekcí struktury písně – přímo nad tabulaturou.

Ukázalo se, že největším problémem analyzovaných aplikací je editace tabulatury. Hlavním ovládacím prvkem analyzovaných aplikací pokrývajících tuto funkcionalitu je klávesnice. Tato klávesnice musí poskytnout spoustu symbolů – noty, značky kytarových technik, čísla pražců, pomlky – a zároveň si zachovat přehlednost. Určit aplikaci, která by nejlépe splnila požadavek 6 nelze. Aplikace, které se snadno ovládaly, neobsahovaly informace o délkách tónů. Naopak hůře ovladatelné aplikace tuto informaci zobrazovaly. Jako nejlepší řešení se nabízí inspirace mechanikou ovládání programu Guitar Pro. Vzhledem

k rozšířenosti programu a zkušenostem uživatelů s ním by poté učení ovládnutí mé aplikace probíhalo rychleji.

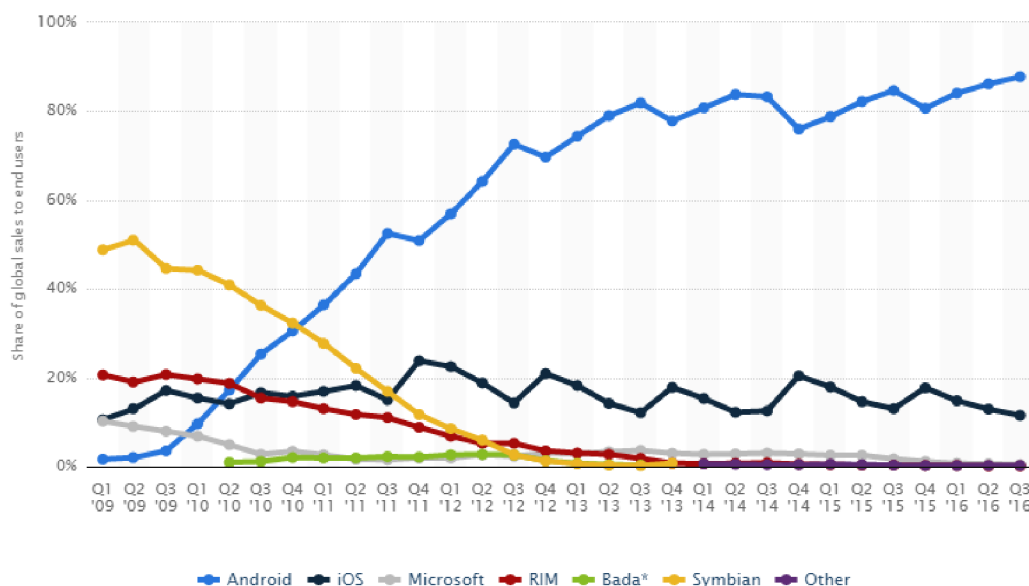
Rozhraní, které by nejvíce vyhovovalo potřebám skupiny InsaneTV má aplikace Song Manager. Hlavní nabídku tvoří seznam skladeb a seznam seznamů skladeb. Ke každé skladbě lze zobrazit strukturu a zaznamenat poznámky k jednotlivým sekcím. Prostor pro poznámky však není dostačující.

Aplikace Lyric Pad poskytuje spoustu pokročilých možností ke správě textů písní. Pro splnění požadavku 5 by však stačil jednoduchý textový editor. Obdobně je tomu tak i u Google kalendáře. Pro splnění požadavku 4 postačí pouze možnost uložení a zobrazení základních informací o koncertech.

## Kapitola 4

# Operační systém Android

Tato kapitola je zaměřena na seznámení s operačním systémem Android, jeho srovnání s ostatními existujícími operačními systémy pro chytrá mobilní zařízení a s jeho API (*Application Programming Interface*), neboli aplikačním programovém rozhraní.



Obrázek 4.1: Ještě v roce 2009 z celkového množství prodaných chytrých telefonů obsahovalo pouze 1.6% Android. S 48.8% trhu vedl operační systém Symbian. Zlom nastal až v roce 2011, kdy se Android dostal na první místo se zastoupením 36.4% <sup>2</sup>.

Jak ukazuje graf na obrázku 4.1, na konci první poloviny roku 2016 z celkového množství prodaných chytrých telefonů fungovalo 86.2% pod operačním systémem Android, což z něj dělá nejrozšířenější mobilní operační systém.

<sup>2</sup>Dostupné z <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>.

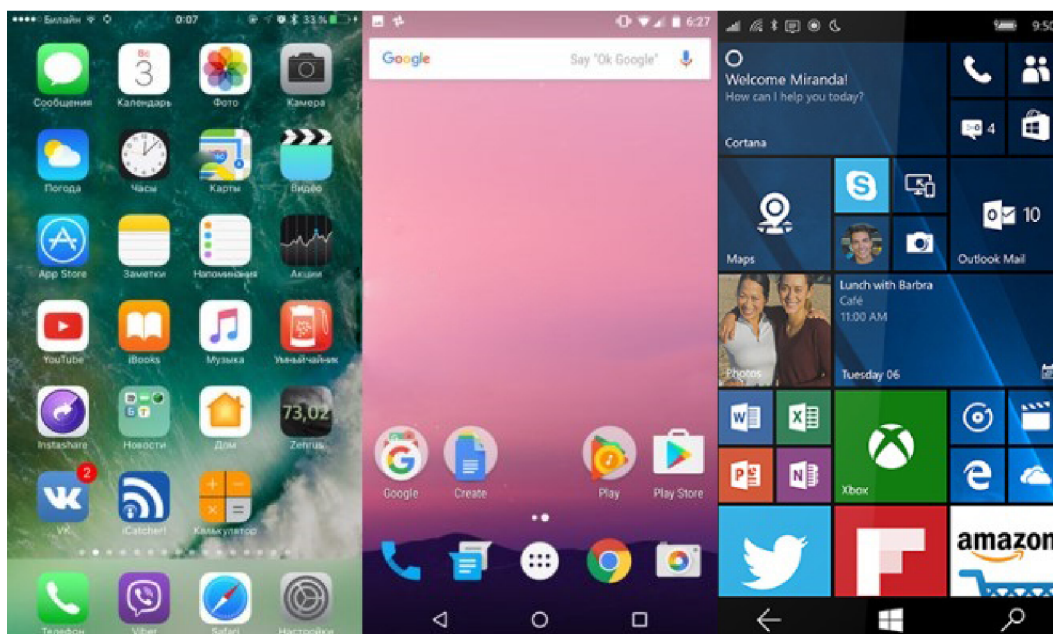
## 4.1 Základní informace

Android je mobilní operační systém vyvíjený pro mobilní zařízení, jako jsou třeba chytré telefony, tablety, navigace, chytré hodinky a podobně. Velká část tohoto operačního systému je dostupná jako otevřený systém (*open source*) pod licenci Apache 2.0<sup>3</sup>, což umožňuje výrobcům zařízení přizpůsobit systém hardwaru a integraci aplikací.

Zajímavostí je, že jednotlivé verze operačního systému Android jsou pojmenované po různých cukrovinkách a sladkostech. V září roku 2016 byla nejrozšířenější verzí KitKat 4.4 s podílem 27.7%. Druhou nejrozšířenější verzí byla Lollipop 5.1 s podílem 21.9%. Dále pak Marshmallow 6.0 s 18.7%, Lollipop 5.0 s 13.1% a Jelly Bean 4.2.x s 7.7% [7]. Ve třetí čtvrtině roku 2016 byla vydána nejnovější verze operačního systému s názvem Nougat 7.0.

## 4.2 Srovnání mobilních operačních systémů

Zaměřím se především na operační systémy s největším podílem na trhu – tedy Android společnosti Google, iOS společnosti Apple a Windows Phone společnosti Microsoft.



Obrázek 4.2: Srovnání hlavních obrazovek operačních systémů. Vlevo iOS 10, uprostřed Android Nougat, vpravo Windows 10<sup>5</sup>.

Android je otevřený systém, společnost Google tedy nepožaduje žádné licenční poplatky za používání této platformy. To má za následek spoustu různých verzí tohoto systému. Základní principy jsou sice vždy stejné, ovšem jednotlivé mechanismy se mohou výrazně odlišovat. V důsledku toho působí uživatelské rozhraní napříč různými zařízeními nekonzistentně. Na nových zařízeních se mohou vyskytovat i dost archaické verze tohoto systému.

<sup>3</sup>Softwarová licence společnosti Apache Software Foundation. Licence umožňuje software svobodně upravovat, distribuovat a licencovat upravenou verzi softwaru. Více na <https://www.apache.org/licenses/LICENSE-2.0>

<sup>5</sup>Dostupné z <http://www.techinja.eu/wp-content/uploads/2016/07/androidvsios.png>.

Na druhou stranu, otevřenost platformy žene její vývoj dopředu. Aplikace nejsou nijak kontrolovány, což má za následek, že existuje nepřehledné množství různých aplikací o různé kvalitě.

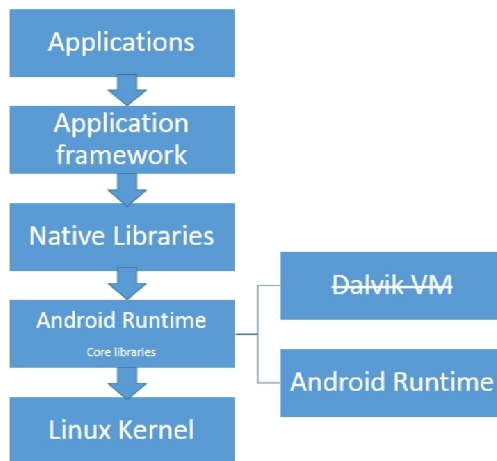
Operační systém iOS společnosti Apple je uzavřeným systémem, to znamená, že nedovoluje ostatním výrobcům využívat tento systém na svých výrobcích. O jeho vývoj se stará výhradně společnost Apple. Díky tomu je uživatelské rozhraní na všech zařízeních konzistentní. Aktualizace na nové verze iOS jsou ihned dostupné všem uživatelům. Apple pečlivě kontroluje a testuje všechny aplikace, což sice může omezovat vývojáře, nicméně uživatelé se mohou spolehnout na funkčnost.

Windows Phone od společnosti Microsoft je také uzavřeným operačním systémem, nicméně Microsoft umožňuje ostatním výrobcům využívat tento operační systém za licenční poplatek. Na trhu existuje zdatelně menší množství zařízení s Windows Phone než s výše zmíněnými operačními systémy. Kvůli tomu pro tuto platformu existuje výrazně menší počet aplikací. Stejně jako Apple, společnost Microsoft aplikace důkladně kontroluje a testuje. Uživatelské rozhraní je specifické svým originálním systémem takzvaných dlaždic.

### 4.3 Architektura

Architektura operačního systému Android se skládá z pěti vrstev [2]. Jednotlivé vrstvy jsou vyobrazeny na obrázku 4.3.

Nejnižší vrstvou architektury je jádro operačního systému, které je postaveno na jádru Linux. Jádro tvoří abstraktní vrstvu mezi hardwarem mobilního zařízení a jeho softwarem [2]. Obsahuje také veškeré ovladače jednotlivých částí zařízení, jakými jsou například fotoaparát, displej, klávesy a podobně. Dále využívá vlastností systému Linux jako je správa paměti, procesů, napájení a mnoho dalších.



Obrázek 4.3: Vrstvy operačního systému Android. Odspodu – jádro, Android runtime, knihovny, aplikační rámeček a aplikace <sup>7</sup>.

Další vrstvy tvoří knihovny, které poskytují základní funkce systému a virtuální stroj s názvem *Android Runtime* (předchůdce byl Dalvik). Knihovny jsou napsány v jazyku C/C++.

<sup>7</sup>Dostupné z [https://www.safaribooksonline.com/library/view/mobile-application-penetration/9781785883378/graphics/B05055\\_02\\_02.jpg](https://www.safaribooksonline.com/library/view/mobile-application-penetration/9781785883378/graphics/B05055_02_02.jpg).

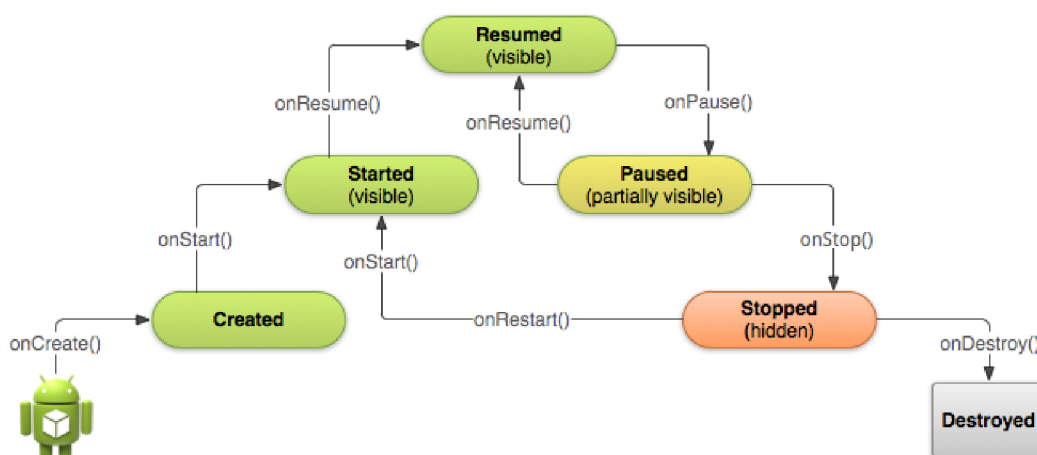
Virtuální stroj slouží pro překlad aplikací z jazyka Java do nativního kódu systému Android. Tato vrstva také obsahuje knihovny specifické pro vývoj aplikací pomocí programovacího jazyka Java [2].

Další vrstvou architektury je aplikační rámec, který poskytuje API pro běh a programování Java aplikací. Tato vrstva je využívána především vývojáři aplikací. Nejvyšší vrstvu tvoří všechny aplikace, příkladem může být internetový prohlížeč, seznam kontaktů.

## 4.4 Android API

Aplikační programové rozhraní určuje jaké třídy a jakou funkcionalitu může vývojář při tvorbě aplikace využít. Před vývojem aplikace musí vývojář stanovit minimální úroveň API. Platí, že čím nižší úroveň API, tím více zařízení bude s aplikací kompatibilní. Ovšem s vyšší úrovní API se vývojáři zpřístupňují pokročilejší funkce a možnosti. Programování se tím stává pohodlnější a snadnější [5].

Důležitým prvkem aplikačního rozhraní jsou aktivity. Jsou to základní komponenty pro vykreslení grafického návrhu aplikace. Úvodní obrazovka aplikace je aktivita, menu je další aktivitou. Je potřeba, aby vývojář ošetřil přepínání mezi aktivitami, ale také různé jiné události, například příchod textové zprávy, příchozí hovor, ukončení aplikace a podobně. Proto existují různé metody, které se volají v různých situacích. Tyto metody popisuje životní cyklus aktivit, který znázorňuje obrázek 4.4.



Obrázek 4.4: Životní cyklus aktivit. Při spuštění aplikace se spustí hlavní aktivita, která se téměř ihned přepíná do stavu *Resumed*, kde zůstává po dobu, kdy je aktivní. Při spuštění další aktivity, která ovšem kompletně nenahradí předchozí aktivitu, se původní aktivita přesune do pozadí a uživatel s ní nemůže nijak manipulovat (stav *Paused*). V tomto stavu jsou zachovány všechny probíhající akce a informace, které jsou důležité pro případ, že se uživatel vrátí do původní aktivity. Ve stavu *Destroyed* jsou uvolněny veškeré zdroje, které aktivita využívala.[9][1].



## 4.5 Material Design

*Material Design* je vizuální jazyk vyvinutý společností Google za účelem vytvoření jednotného grafického systému, který uživateli poskytuje stejnou zkušenost napříč různými platformami a velikostmi zařízení [6]. *Material Design* popisuje jak vzhled samotných prvků, tak i celek – tedy vzdálenosti mezi jednotlivými prvky, animace, apod. Základní stavební kameny jazyku – světlo, povrch a pohyb – uživateli sdělují, jak se objekty pohybují, interagují a existují jak v prostoru, tak mezi sebou navzájem. Prvky se nachází v trojrozměrném prostoru. Jejich umístění je dáno osami  $x$ ,  $y$ ,  $z$ , přičemž osa  $z$  určuje výšku prvku. Tato výška je pak uživateli znázorněna stínem pod prvkem. Výška také určuje pořadí vykreslení prvků. Uživatel může interagovat pouze s prvkem, který je nejvýše. *Material Design* byl poprvé představen ve verzi 5.0 Lollipop, ovšem díky knihovnam je možné jej používat i ve starších verzích.

## 4.6 Dostupné nástroje pro vývoj

Oficiální nástroj pro vývoj aplikací pro operační systém Android je Android Studio [3]. Podporuje navrhovat design buď v XML nebo ve speciálním návrhovém režimu, který umožňuje jednoduše přesouvat prvky přímo na obrazovku namísto jejich psaní v XML. Emulátory podporují několik předdefinovaných zařízení, ovšem lze vytvořit i vlastní zařízení, u kterých lze vybrat verzi API, velikost RAM, rozlišení, kamery a mnoho dalších parametrů.

Dalším vývojovým prostředím je NetBeans. Tam je vývoj aplikací pro Android umožněn instalací pluginu NBAndroid. Veškeré knihovny je však potřeba nainstalovat zvlášť. NetBeans také nativně nepodporuje náhled rozvržených prvků při implementaci. Naštěstí existují pluginy, které dělají vývoj v tomto nástroji daleko příjemnější.

Dalším oblíbeným a rozšířeným prostředím pro vývoj je Eclipse, který byl ve spojení s pluginem ADT<sup>8</sup> oficiálním nástrojem pro vývoj aplikací před nástupem Android Studia [3]. Stejně jako Android Studio, Eclipse podporuje implementaci uživatelského rozhraní buď v XML nebo v návrhovém režimu.

---

<sup>8</sup>Android Development Tools

## Kapitola 5

# Návrh aplikace

Tato kapitola pojednává o procesu návrhu aplikace. Nejprve se zaměří na návrh konceptuálního modelu aplikace na základě požadavků z kapitoly 3. Poté na návrh samotného uživatelského rozhraní, včetně prvotních náskresů. Nechybí ani výběr názvu aplikace.

### 5.1 Cílová skupina uživatelů

I přestože požadavky na aplikaci byly specifikovány na základě potřeb hudební skupiny InsaneTV, jsou použitelné i pro potřeby jiných kapel. Cílovou skupinou uživatelů jsou všichni hudebníci, kteří si chtějí zjednodušit a zefektivnit správu nad tvorbou jejich hudební skupiny. U uživatelů se předpokládají alespoň minimální znalosti hudební teorie popsané v kapitole 2.

Jako minimální úroveň API jsem zvolil úroveň 16. Aplikace je tedy kompatibilní se zařízeními s verzí Androidu 4.2.x a výše, což pokryje přibližně 95,2% uživatelů operačního systému Android [4]. Android může fungovat na mnoha různých zařízeních s rozdílným rozlišením displeje a výkonem hardwaru. S tím je nutné při návrhu počítat.

### 5.2 Výběr názvu aplikace

Výběr názvu aplikace není radno podcenit. Je to právě název společně s logem co na první pohled potenciálního uživatele zaujme v seznamu vyhledávaných aplikací. Název aplikace by proto měl zaujmout a nějakým způsobem vybočovat mezi ostatními názvy. Ideálně by měl také napovědět, k jakému účelu aplikace slouží.

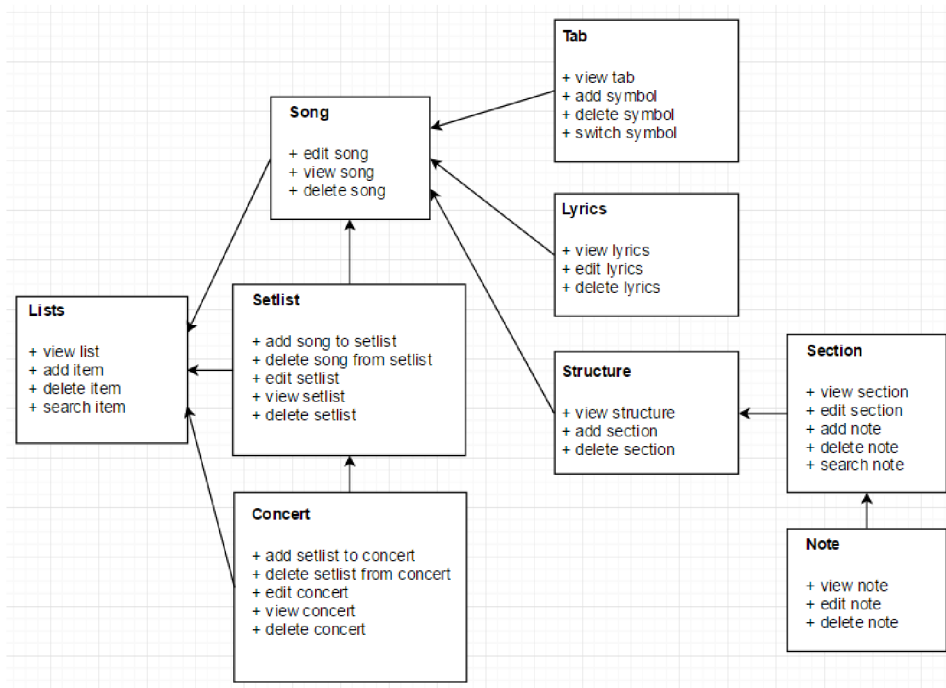
Pro mou aplikaci sloužící ke správě tvorby hudební skupiny jsem zvolil jméno MyBand-Song. Název je krátký a zároveň vystihuje funkcionalitu aplikace. Název také reflektuje, že aplikace slouží ke správě skladeb uživateli hudební skupiny a ne ke správě hudebních souborů uložených v telefonu.



Obrázek 5.1: Logo aplikace. Zvoleny jsou výrazné kontrastní barvy, které upoutají pozornost.

## 5.3 Konceptuální model

Konceptuální model specifikuje všechny objekty a akce, se kterými je uživatel v přímé interakci a jsou pro něj přímo viditelné. Proto je dobré se při návrhu řídit pravidlem, že co není v konceptuálním modelu, o tom by uživatel neměl vědět [10].



Obrázek 5.2: Konceptuální model k aplikaci MyBandSong.

Vytvořený konceptuální model k mé aplikaci je vidět na obrázku 5.2. Hlavní objekt modelu, *Lists*, si lze představit jako kontejner, ve kterém jsou uloženy tři šanony. V jednom jsou uchovány všechny papíry s informacemi o písních. Ve druhém jsou papíry se seznamy skladeb a ve třetím papíry s informacemi o koncertech. Toto uspořádání minimalizuje chaos, který vládne organizaci poznámek skupiny.

Ke každé položce písně je kromě základních informací o ní k dispozici také její struktura, text a tabulatura. Struktura písně je složena ze seznamu sekcí. Každá sekce pak organizuje různé poznámky, které kapela potřebuje uchovat (viz požadavek 7).

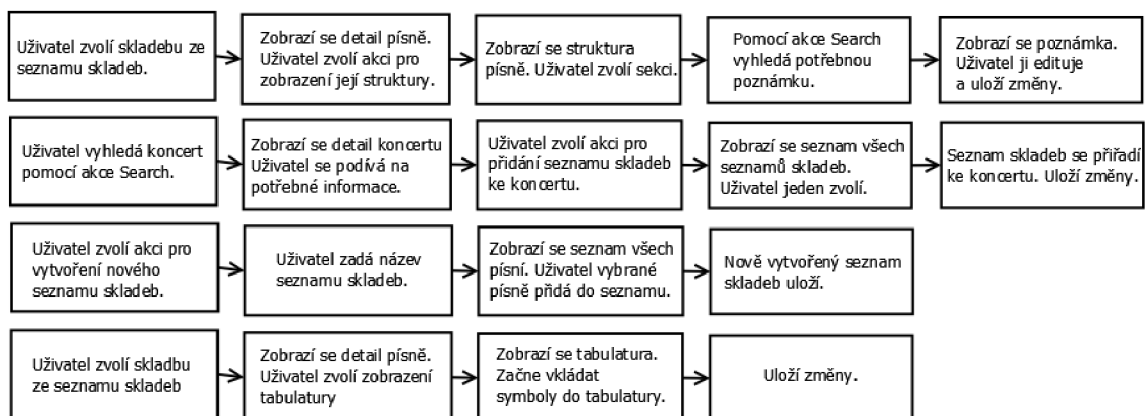
### 5.3.1 Akce

Správné pojmenování operací, které se mohou nad konceptuálním modelem provádět, ovlivňuje jak rychle se uživatelem pečlivě kontrolované a pomalé úkony stávají úkony rychlé a automatické [10]. Jak uvádí Johnson [10], terminologie operací modelu je zaměřena na úkony, což uživateli usnadňuje orientaci v nich. Akce konceptuálního modelu se dají rozdělit do obecnějších kategorií podle typu, a to na *Add*, *Delete*, *Edit*, *View*, *Search* a *Switch*. To značí, že jejich pojmenování je konzistentní. Důsledkem je, že se uživatel nemusí učit spoustu různých názvů, ale jen pár, které jsou využívány napříč celým systémem.

Objekty, které jsou seznamem, potřebují akce pro přidání a odebrání položky. Některé seznamy budou rozsáhlejší než jiné, z toho důvodu jsou u některých objektů navíc akce

pro vyhledání položky. Vzhledem k tomu, že se struktura písně skládá z chronologicky seřazených sekcí, orientace zde bude jednoduchá. Vyhledávání není potřeba ani v seznamu skladeb. Předpokladem totiž je, že seznamy skladeb mají setřídít a seskupit repertoár kapely. Neměly by tedy obsahovat příliš velké množství písní.

Na obrázku 5.2 je také vidět, že akci editace nemají všechny objekty. Je přítomna pouze u objektů, které uchovávají informace o sobě samém. Příkladem může být text poznámky nebo jméno písně. Nejčastější akce nad modelem, které vychází z pozorování skupiny popsaného v kapitole 3 zobrazují scénáře na obrázku 5.3.



Obrázek 5.3: Scénáře popisující uživatelskou interakci s modelem. První řádek: uživatel chce editovat poznámku v sekci struktury. Druhý řádek: uživatel si chce prohlédnout informace o koncertu a přiřadit k němu seznam skladeb. Třetí řádek: uživatel chce vytvořit seznam skladeb. Čtvrtý řádek: uživatel chce zapsat hudební nápad.

### 5.3.2 Matice objektů a akcí

Pro lepší znázornění komplexnosti návrhu konceptuálního modelu jsem použil matici objektů a akcí, kterou vyobrazuje následující tabulka 5.1.

	Add	Delete	Edit	View	Search	Switch
<b>Tab</b>	X	X	X	X		X
<b>Structure</b>	X	X		X	X	
<b>Lists</b>	X	X		X	X	
<b>Section</b>	X	X	X	X		
<b>Concert</b>	X	X	X	X		
<b>Setlist</b>	X	X	X	X		
<b>Song</b>	X	X	X	X		
<b>Lyrics</b>		X	X	X		
<b>Note</b>		X	X	X		

Tabulka 5.1: Matice objektů a akcí k mému konceptuálnímu modelu. Popisy sloupců tvoří akce, popisy řádků zase objekty. Křížky je vyznačeno, jaké akce lze provádět nad jakým objektem.

Malá a celkem hustě vyplněná matice indikuje, že operace nad jednotlivými objekty se často opakují. Podle Johnsona [10] se touto vlastností vyznačuje konzistentní návrh. Uživatel se tedy nebude muset příliš dlouho učit význam spousty různých operací. Matice je však trochu delší, takže uživatel bude muset zvládnout práci s více koncepty.

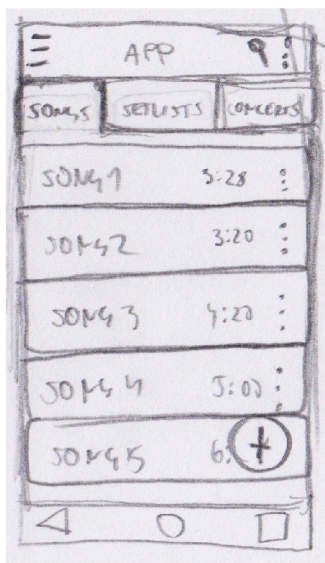
Matice také ukazuje, že objekty stejného datového typu sdílejí stejné, nebo alespoň podobné operace, což je také znamením dobrého návrhu [10]. Většina objektů jsou seznamy. *Lists* obsahuje seznam objektů, stejně tak jako *Setlist* nebo *Concert*. Seznamem je ale také struktura (seznam sekcí) a sekce (seznam poznámek). Poznámky a text skladby jsou dalším datovým typem.

## 5.4 Uživatelské rozhraní

Nyní následuje návrh samotného uživatelského rozhraní aplikace. Popsány jsou především klíčové části aplikace, ke kterým je přiložena i skica.

### 5.4.1 Hlavní obrazovka

Hlavní obrazovka tvoří nejdůležitější část aplikace. Slouží jako brána k pokročilejším funkcím aplikace. Jak je vidět na obrázku 5.4, hlavní obrazovka odpovídá kontejneru z konceptuálního modelu, akorát zde jsou šanony nahrazeny třemi seznamy a papíry položkami v nich (viz popis obrázku 5.2). Z této obrazovky bude možné provádět nad položkami veškerou organizaci, jako je například jejich přidávání, mazání, změna názvu.

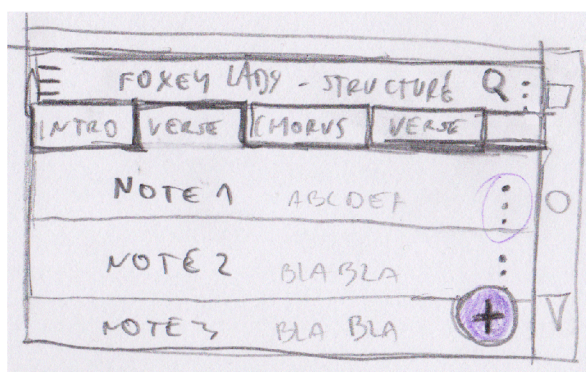


Obrázek 5.4: Hlavní obrazovka aplikace zobrazuje tři seznamy mezi kterými lze přepínat. První a základní je seznam skladeb. Druhý je seznam seznamů skladeb. Třetí seznam koncertů. Vzhledem k tomu, že se jedná o seznamy všech vytvořených položek, nesmí chybět možnost vyhledání položky podle konkrétního názvu. K tomu bude po vzoru hudebních přehrávačů sloužit ikona lupy v pravém horním rohu.

### 5.4.2 Struktura písně

U návrhu zobrazení struktury jsem se potýkal s problémem, jak na jedné obrazovce zobrazit současně seznam sekcí a ke každé sekci ještě seznam poznámek. Z obrázku 5.5 lze usoudit, že návrh je pojat podobným stylem, jako je tomu u hlavní obrazovky. Navigační lišta tentokrát slouží k zobrazení sekcí struktury. Ovšem s tím rozdílem, že zde v ní není počet prvků pevně dán – sekce lze přidávat. Kvůli zachování konzistence se jako u ostatních seznamů přidá nově vytvořená sekce na konec. Uživatel bude mít možnost sekci přesunout.

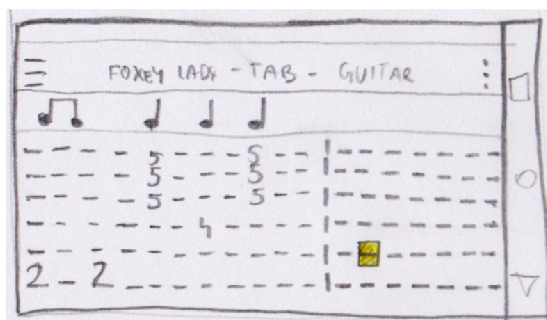
Poznámek u každé sekce může být mnoho. Proto zde bude k dispozici vyhledávání podle názvu. U každé poznámky bude také kromě názvu zobrazen i krátký výňatek jejího textu, což usnadní vyhledávání pohledem.



Obrázek 5.5: Struktura písně společně s poznámkami u vybrané sekce.

### 5.4.3 Tabulatura

Zobrazení tabulatury jsem se rozhodl pojat formou pokročilejšího textového editoru. Jak je vidět na obrázku 5.6, notové linky zde reprezentují řádky pomlček. Při zápisu hudby symbol tuto pomlčku nahradí, proto je důležité, aby písmo bylo neproporciální<sup>1</sup>, čímž se zachovají pozice znaků. Pro čitelnější vizualizaci délky tónu se s narůstající délkou bude zvyšovat i počet pomlček, které odsazují další symbol.

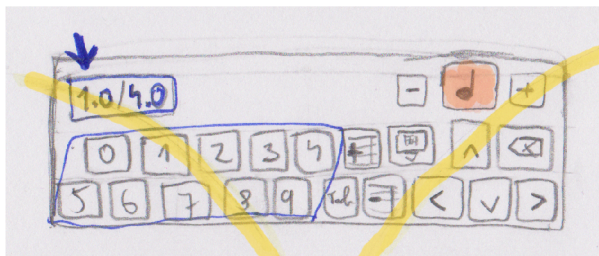


Obrázek 5.6: Tabulatura s vyobrazenou délkou tónů. Jak již bylo řečeno v kapitole 2, informace o délkách tónů se píše nad tabulaturu. Uživatelem zvolená pozice je zvýrazněná.

<sup>1</sup>Písmo, v němž mají všechny znaky stejnou šířku.

#### 5.4.4 Zápisy hudby

Zápis hudby do tabulatury se v analýze existujících řešení ukázal jako nejvíce kritický bod. Stejně jako u ostatních aplikací, i zde bude k zápisu symbolů sloužit klávesnice, která je vyobrazena na obrázku 5.7.



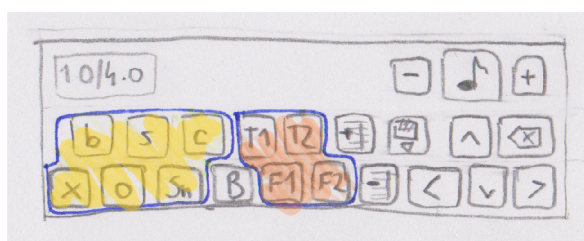
Obrázek 5.7: Skica klávesnice sloužící k zápisu symbolů do tabulatury.

Klávesy jsou rozmístěny tak, aby byly v dosahu přirozené dráhy pohybu prstů nebo ruk (naznačeny žlutými čarami). Velikost kláves byla zvolena tak, aby minimalizovala možnost přehmatu.

Stejně tak bylo zvoleno i jejich rozmístění. Pravá polovina klávesnice obsahuje funkcionální klávesy, levá číselné. Číselné klávesy zabírají nadpoloviční velikost klávesnice, aby se na ně bylo možné rychle přesunout z obsluhy funkcionálních kláves a při psaní symbolů tím pádem využívat obou ruk, případně prstů. Funkcionální klávesy, které se nebudou používat tak často, jsou umístěny více doprostřed klávesnice.

Pro snadnější úpravu tabulatury je rovněž vhodné zobrazit, kolik dob již bylo do taktu vloženo a kolik dob ještě zbývá, aby byl takt zaplněn. Tuto hodnotu zobrazuje komponenta, která je na obrázku 5.7 označena modrou šipkou.

Symbol noty (označen oranžově) zobrazuje délku vkládaného tónu. Pro jeho změnu slouží klávesy plus a mínus po jeho stranách. Pro lepší manipulaci s klávesnicí jsou tyto klávesy umístěny na pravé straně. Díky tomu bude uživatel moci levou rukou vkládat symboly a pravou rukou měnit jejich délku.



Obrázek 5.8: Skica klávesnice pro zápis bicích.

Na obrázku 5.8 je zobrazena skica klávesnice pro zápis bicích. Rozmístění obecných ovládacích prvků je stejné jako na předchozí klávesnici (obrázek 5.7). Tabulatura pro označení bicích komponent však nevyužívá čísel, nýbrž písmenných zkratk. Bylo tedy potřeba vymyslet, jak co nejpřehledněji na klávesnici vyobrazit toto velké množství symbolů.

Umístění symbolů na klávesnici je rozvrhnut tak, aby svou pozicí co nejvíce odpovídalo pozici na reálných bicích. Oranžově je na obrázku 5.8 označena skupina komponent,

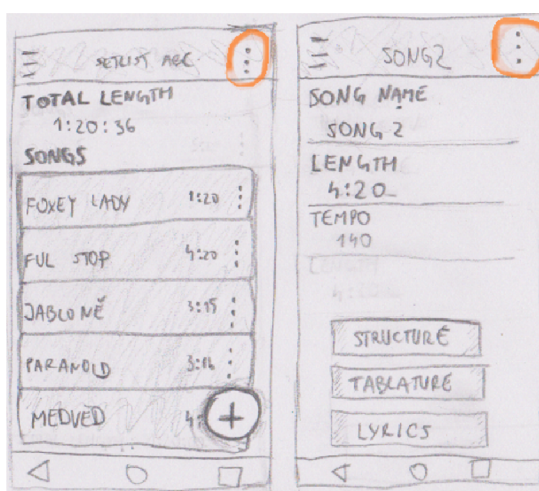
kteře jsou na bicích umístěny více na pravé straně, naopak žlutě jsou označeny komponenty z levé strany. Uprostřed je basový buben. Ve vrchní řadě jsou umístěny komponenty, které se nachází na přední straně bicích. Toto rozmístění však může na první pohled působit zmatečně, proto v aplikaci nesmí chybět nápověda. Srovnání rozmístění symbolů na klávesnici a skutečných bicích vyobrazuje obrázek 5.9.



Obrázek 5.9: Reálné bicí s vyznačenými symboly z klávesnice. Největší rozdíl spočívá v tom, že činely mohou být po všech stranách bicích. Pro větší přehlednost jsem je však na klávesnici umístil na levou vrchní stranu. Přechodové bubny (označené symboly T) se ve větším množství skládají směrem doprava, zde je pouze jeden.

#### 5.4.5 Zobrazení detailů položek

Jak lze vidět na obrázku 5.10, prvky jsou rozmístěny tak, aby ve vrchní části obrazovky byla statická data a ve spodní části prvky, se kterými může uživatel nějak manipulovat. Statickými daty se rozumí názvy a jiné důležité informace, které by měly jako první zaujmout uživatele pozornost.

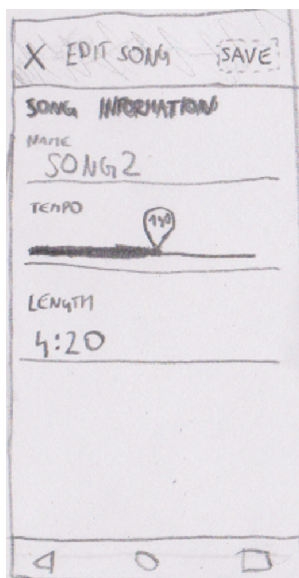


Obrázek 5.10: Návrh zobrazení položky. Vlevo seznam skladeb, vpravo skladba.



#### 5.4.6 Editační formuláře

Editace a další akce nad položkami budou dostupné z kontextového menu, které se zobrazí stisknutím symbolu tří teček (na obrázku 5.10 označeno oranžově). Pro zachování konzistence jsou prvky v editačním formuláři rozmístěny stejně jako v detailu položky. Pro zrychlení editace bude formulář předvyplněn aktuálními hodnotami. Příklad editačního formuláře je na obrázku 5.11.

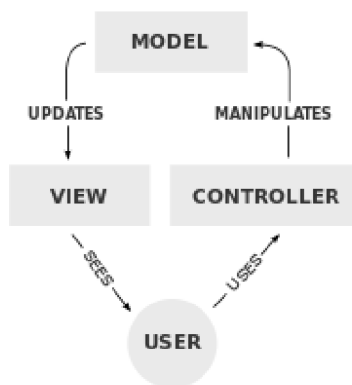


Obrázek 5.11: Editační formulář k položce skladby.

# Kapitola 6

## Implementace

Tato kapitola se zabývá samotnou implementací aplikace. Popsány jsou hlavně netriviální a zajímavé části programu. Největší prostor je věnován implementaci tabulatury, která tvoří stěžejní část aplikace. Snímky hlavních obrazovek jsou k nahlédnutí v příloze B.



Obrázek 6.1: Architektonický vzor *MVC*. Uživatel zadává podněty řadiči (controller), ten přistupuje k modelu a provádí v něm změny. Ty se následně promítnou v zobrazení (view) [12]<sup>2</sup>.

Architektura aplikace nejvíce odpovídá architektonickému vzoru *MVC*<sup>3</sup>, jehož hlavní myšlenkou je oddělení logiky od výstupu. Dělí se na tři komponenty – *model* (soubory Java), *view* (soubory XML) a *controller* (aktivity, fragmenty). Od klasického MVC se však liší tím, že vzhled některých složitějších prvků je definován programově v komponentě *controller*. Obrázek 6.1 vyobrazuje komunikaci mezi jednotlivými komponenty.

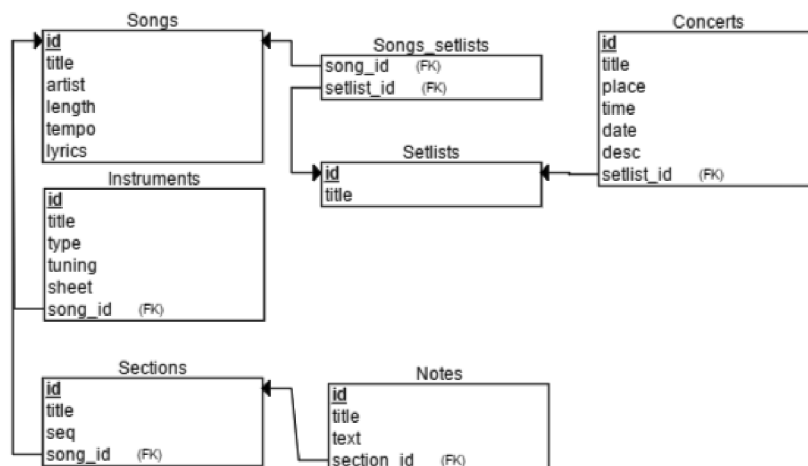
### 6.1 Databáze

K ukládání perzistentních dat aplikace využívá databázového systému *SQLite*. Relační model je vyobrazen na obrázku 6.2. Jako přístupový bod k vytvoření a správě databáze slouží třída *BandDbHelper*, rozšiřující třídu *SQLiteOpenHelper*<sup>4</sup>. Ta poskytuje metody,

<sup>2</sup>Dostupné z <https://en.wikipedia.org/wiki/Model-view-controller>.

<sup>3</sup>Model – View – Controller

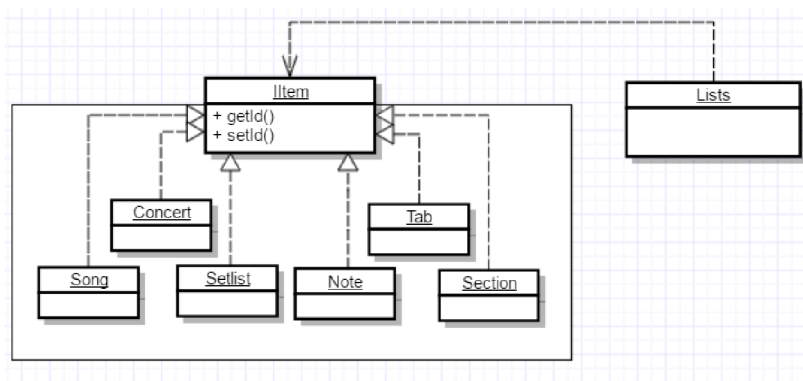
<sup>4</sup>Viz <https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html>.



Obrázek 6.2: Vztahy mezi tabulkami databáze. Schéma koresponduje s konceptuálním modelem z kapitoly 5.

které asynchronně spouští dotazy pro přidání, editaci nebo smazání řádku tabulky databáze. Dále také poskytuje metody, které fungují jako továrny pro vytvoření objektové reprezentace databázových entit. Vstupem těchto metod je *kurzor* obsahující výsledek dotazu, výstupem pak objekt reprezentující entitu v databázi.

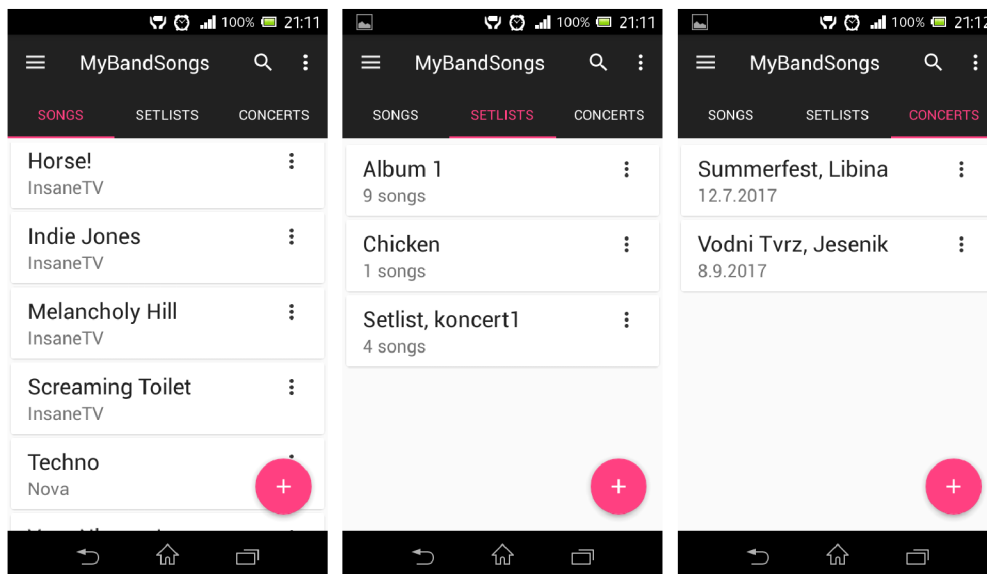
Lokální reprezentace databázových entit využívá návrhového vzoru *fasáda*, kdy se k jednotlivým entitám přistupuje skrze rozhraní (viz obrázek 6.3). Pro správu těchto entit existuje pomocná třída *Lists* obsahující jejich seznamy. Při dotazech do databáze se výsledné položky ukládají do těchto seznamů, což minimalizuje počet opakujících se dotazů na stejné řádky tabulky.



Obrázek 6.3: Lokální reprezentace databázových entit. Pro jednoduchost nejsou v diagramu zahrnuty vztahy mezi entitami.

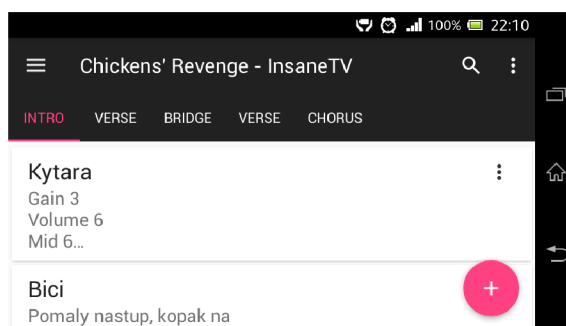
## 6.2 Hlavní obrazovka a obrazovka struktury písňe

Jak je vidět na obrázku 6.4, hlavní obrazovka aplikace se skládá ze tří záložek z nichž každá zobrazuje seznam položek – písňe, setlisty a koncerty. Je implementována aktivitou `MainActivity`. Pro zobrazení záložek je použit widget `TabLayout`<sup>5</sup> ve spojení s elementem `ViewPager`<sup>6</sup>, který umožňuje přepínat mezi stránkami pomocí gesta. Jednotlivé stránky jsou implementovány třídou `Fragment`<sup>7</sup>.



Obrázek 6.4: Hlavní obrazovka. Vlevo záložka se seznamem všech písňi, uprostřed se seznamem seznamů skladeb, vpravo se seznamem koncertů.

Pro přidání nové položky se na obrazovce nachází plovoucí akční tlačítko, po jehož stisknutí se zobrazí formulář. Ten se liší dle typu položek, které jsou zobrazeny v aktuálně zvoleném fragmentu. Obsluha stisknutého tlačítka se deleguje právě do něj.



Obrázek 6.5: Obrazovka se strukturou písňe v horizontálním zobrazení.

<sup>5</sup>Viz <https://developer.android.com/reference/android/support/design/widget/TabLayout.html>.

<sup>6</sup>Viz <https://developer.android.com/reference/android/support/v4/view/ViewPager.html>.

<sup>7</sup>Viz <https://developer.android.com/guide/components/fragments.html>.

Podobným způsobem jako hlavní aktivita je implementována i aktivita pro zobrazení struktury skladby (viz obrázek 6.5). Pro každou sekci struktury existuje odpovídající záložka, poznámky v sekci jsou pak zobrazeny ve fragmentu. Jediný rozdíl je v tom, že zde není počet záložek pevně dán. Lze je dynamicky přidávat a odebírat.

## 6.3 Seznamy hudebních položek

Aplikace často pracuje se seznamy, proto bylo nutné najít způsob, jak je vhodně zobrazit. Nejlepší možnosti poskytuje widget `RecyclerView`<sup>8</sup>, díky kterému je možné zobrazit i velký počet dat. Načítá totiž pouze zobrazení prvků, které se vlezou na obrazovku.

Položky seznamu v sobě spravují objekty zvané *adaptéry*, které rozšiřují třídu `RecyclerView.Adapter`<sup>9</sup>. Adaptéry implementované v aplikaci `MyBandSong` mimo jiné také poskytují operace nad položkami seznamu (editace, mazání, apod.).

V některých případech je po provedení změn nad položkou seznamu potřeba explicitně upozornit aktivitu, která tyto změny v adaptéru vyvolala. Taková aktivita musí implementovat vytvořené rozhraní `IResponder`, které vychází z návrhového vzoru *pozorovatel*. Této funkcionality je potřeba například v detailu setlistu kde je vyobrazen i seznam přiřazených skladeb. Při editaci některé skladby adaptér upraví pouze patřičný prvek seznamu. Je však potřeba také přepočítat délku seznamu skladeb v samotném objektu setlist, ke kterému adaptér nemá přístup, a poté aktualizovat celé uživatelské rozhraní.

## 6.4 Poznámky a text skladby

V aktivitách vyobrazujících text skladby nebo text poznámky je pro lepší čitelnost umožněno uživateli gestem zvětšit nebo zmenšit velikost fontu. K detekci takovýchto gest se používá třída `ScaleGestureDetector`<sup>10</sup>. Ta pak při detekování gesta spustí algoritmus, kterým se změní velikost písma. Princip algoritmu vyobrazuje ukázka 6.1.

---

```
1 aktualniVelikost = text.ziskejVelikost();
2 faktorZvetseni = aktualniVzdalenostMeziDvemaPrsty / puvodniVzdalenostMeziDvemaPrsty;
3 novaVelikost = aktualniVelikost * faktorZvetseni;
4
5 novaVelikost = orezVelikostPodleLimitu(novaVelikost);
```

---

Listing 6.1: Pseudokód, který popisuje, jakým způsobem se při detekci gesta mění velikost fontu.

## 6.5 Přidání a editace položky

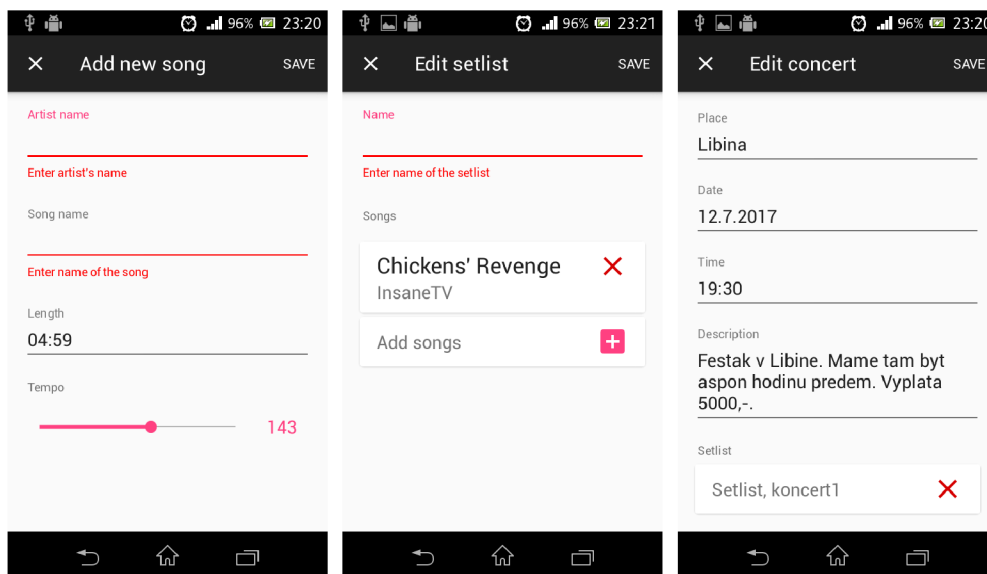
Pro implementaci formuláře pro přidání nebo editaci položky je použit návrhový vzor *strategie*. Aktivita implementuje obecný formulář, jehož funkcionality se odvíjí od toho, zda se mu předá identifikátor položky nebo ne. Pokud se předá, jedná se o editační formulář. Podle identifikátoru se získá odpovídající položka a následně se na základě jejích hodnot předvyplní formulář. V opačném případě zůstává formulář prázdný a po jeho odeslání se vytváří položka nová. Některé z formulářů v aplikaci vyobrazuje obrázek 6.6.

---

<sup>8</sup>Viz <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html>.

<sup>9</sup>Viz <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.Adapter.html>.

<sup>10</sup>Viz <https://developer.android.com/reference/android/view/ScaleGestureDetector.html>.



Obrázek 6.6: Formuláře. Vlevo formulář pro přidání nové písně (s chybovou hláškou), uprostřed formulář pro editaci seznamu skladeb (také s chybovou hláškou), vpravo formulář pro editaci koncertu.

Po stisku tlačítka pro uložení se nejprve provede validace vyplněných hodnot. Pokud je formulář chybně vyplněn, zobrazí se patřičné chybové hlášky. V opačném případě se vyplněné hodnoty vloží do objektu typu `Intent`<sup>11</sup>, který je poté předán do původní aktivity skrz metodu `onActivityResult`. Původní aktivita, ze které se aktivita s formulářem spustila, pak objekt předá do adaptéru, který na jeho základě aktualizuje uživatelské rozhraní.

## 6.6 Tabulatura

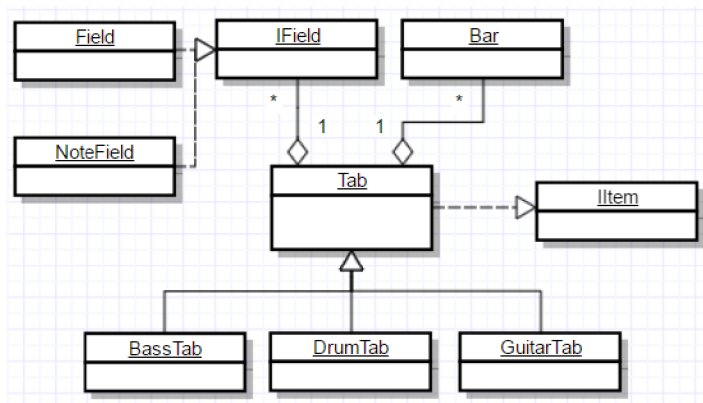
Aktivita tabulatury tvoří nejsložitější část aplikace. Cílem bylo vytvořit jednoduchý a intuitivní editor tabulatur, který bude umožňovat zaznamenat hudební nápady nebo celé písně. Mezi největší problémy, které jsem při implementaci editoru musel řešit, spadá optimalizace a vykreslování správně spojených not.

### 6.6.1 Model

Model tabulatury využívá návrhového vzoru *šablonová metoda*. Hlavní mechaniky pro operace nad tabulaturou obsahuje abstraktní třída `Tab`, kterou rozšiřují další třídy reprezentující konkrétní hudební nástroje. Díky tomu jsou otevřeny dveře pro snadnou rozšiřitelnost o další typy nástrojů. Diagram tříd modelu tabulatury zobrazuje obrázek 6.7.

Pole na tabulatuře je symbolizováno rozhraním `IField`, které implementují třídy `Field` (symboly tabulatury) a `NoteField` (ikony not nad tabulaturou). Takty reprezentuje třída `Bar`, která rozděljuje políčka do logických celků. Do databáze se tabulatura ukládá ve formě textového řetězce. Toho je dosaženo použitím serializace pomocí konečného automatu.

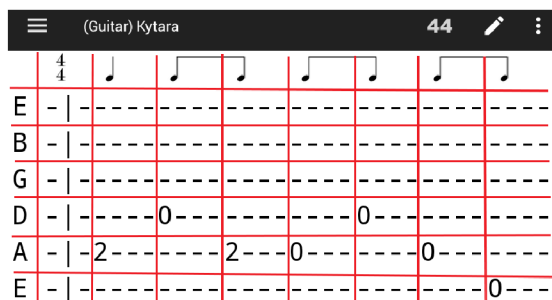
<sup>11</sup>Viz <https://developer.android.com/reference/android/content/Intent.html>.



Obrázek 6.7: Diagram tříd modelu tabulatury. Pro jednoduchost nejsou zobrazeny metody a atributy. Konkrétní nástroje reprezentují třídy `GuitarTab`, `BassTab` a `DrumTab`.

## 6.6.2 Zobrazení

Jednotlivé pole tabulatury jsou zobrazeny v horizontálně orientovaném widgetu `RecyclerView` ve spojení s widgetem `GridLayoutManager`<sup>12</sup>, který umožňuje uspořádat prvky seznamu do mřížky (viz obrázek 6.8). Položky seznamu tvoří widgety `TextView`<sup>13</sup>.



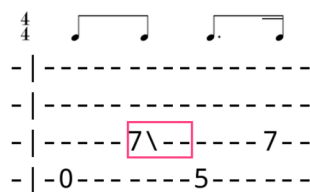
Obrázek 6.8: Zobrazení prvků seznamu v mřížce. Červené čáry vizualizují hranice prvků `TextView`, které vyobrazují symboly tabulatury.

Jak již bylo zmíněno v sekci 6.3, widget `RecyclerView` vykresluje pouze položky, které se vejdou na obrazovku. Díky tomu není potřeba vykreslovat všechny prvky tabulatury, což by aplikaci značně zpomalilo. Kdyby však byl každý znak tabulatury reprezentován jedním elementem, byl by počet zobrazených prvků stále vysoký. Proto bylo nutné jejich počet zredukovat na nutné minimum. Toho bylo dosaženo tak, že jeden element zobrazuje jeden celý symbol včetně technik a odsazení pomlčkami (viz obrázek 6.9).

Jak je vidět na obrázku 6.9, ikony not jsou umístěny nad symbolem tabulatury. Vzhledem k tomu, že pro různé velikosti displeje se mění i velikost ikon, bylo potřeba nastavit pro různá rozlišení správné velikosti fontů. Obrazovky se rozdělují do šesti typů na základě

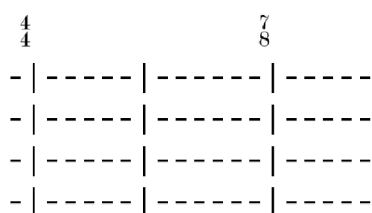
<sup>12</sup>Viz <https://developer.android.com/reference/android/support/v7/widget/GridLayoutManager.html>.

<sup>13</sup>Viz <https://developer.android.com/reference/android/widget/TextView.html>.



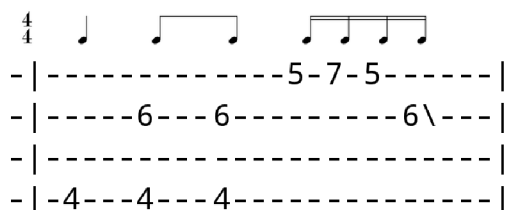
Obrázek 6.9: Růžově ohraničená oblast reprezentuje jeden element.

jejich denzity<sup>14</sup>. Nejprve jsem ručním zkoušením našel správnou velikost fontu pro jednu denzitu. Pro ty další pak byly velikosti vypočteny na základě již nalezené hodnoty a rozlišení ikony pro danou denzitu.



Obrázek 6.10: Označení taktů pouze při změně metra.

Při vykreslování ikon se nejprve na základě pozice vykreslovaného prvku zjistí, zda neodpovídá začátku taktu. Pokud ano, je potřeba nastavit ikonu metra. Jak je vidět na obrázku 6.10, označení taktu se vykreslí pouze při změně metra, což zlepšuje přehlednost tabulatury.



Obrázek 6.11: Spojení not pomocí trámců v čtyřčtvrtovém taktu. Délka základní doby je zde jedna.

Správná ikona noty se vybírá na základě délky doby. Problém nastává u not, jejichž délka je menší než jedna (osminové, šestnáctinové, dvaatřicetinové). Ty se pro lepší vizuální orientaci v počtu dob spojují trámci (viz obrázek 6.11). Jak bylo zmíněno v kapitole 1, noty se spojují po délce základní doby taktu. Pro každé pole, které má zobrazit takovou ikonu se spustí algoritmus, který od začátku taktu hledá skupiny těchto not. Pokud aktuální nota spadá do takové skupiny, tak na základě délky noty předcházející a následující vybere odpovídající ikonu. Při každé změně je pak nutné znovu vykreslit celý takt, protože se skupiny spojovaných not mohou změnit.

<sup>14</sup>Viz [https://developer.android.com/guide/practices/screens\\_support.html](https://developer.android.com/guide/practices/screens_support.html).



### 6.6.3 Ovládání

Definice vzhledu klávesnice je rozdělena do dvou částí. Spodní část (na obrázku 6.12 část s tlačítky) je popsána ve vlastním souboru XML pomocí rozvržení `Keyboard`<sup>15</sup>. Klávesnici pak na obrazovce vykresluje widget `KeyboardView`<sup>16</sup>, který zároveň detekuje stisknutí kláves.



Obrázek 6.12: Klávesnice pro zápis symbolů. Spodní část obsahuje tlačítka, horní část prvky informačního charakteru a tlačítka pro změnu délky vkládaného symbolu.

Horní část klávesnice je definována ve vlastním pohledu, protože využívá prvků, které nejsou v rozvržení `Keyboard` podporovány. Jak je vidět na obrázku 6.12, tato část se skládá z jednoho řádku. Tlačítka jsou implementována pomocí widgetu `ImageButton`<sup>17</sup>, ikona mezi nimi pak pomocí `ImageView`<sup>18</sup>. Elementy jsou zasazené do rozvržení `TableLayout`<sup>19</sup>, aby byly zobrazeny v jednom řádku.

Obsluhu kláves zajišťuje vytvořená třída `CustomKeyboard`, která na základě kódu stisknutých kláves spouští příslušné operace. Kvůli minimalizaci zátěže hlavního vlákna probíhají tyto operace asynchronně.

<sup>15</sup>Viz <https://developer.android.com/reference/android/inputmethodservice/Keyboard.html>.

<sup>16</sup>Viz <https://developer.android.com/reference/android/inputmethodservice/KeyboardView.html>.

<sup>17</sup>Viz <https://developer.android.com/reference/android/widget/ImageButton.html>.

<sup>18</sup>Viz <https://developer.android.com/reference/android/widget/ImageView.html>.

<sup>19</sup>Viz <https://developer.android.com/reference/android/widget/TableLayout.html>.

# Kapitola 7

## Testování

Nedílnou součástí vývoje aplikace je i její testování. Právě to odhalí nejvíce chyb jak ve funkcionalitě, tak v návrhu. V této kapitole jsou popsány jednotlivé kroky testování a zhodnocení výsledků.

### 7.1 Optimalizace

Jak bylo psáno v kapitole 4, mobilní zařízení mohou fungovat pod různými verzemi systému Android. Bylo tedy důležité aplikaci optimalizovat pro co nejširší okruh zařízení. Dále bylo nutné otestovat, zda rozvržení prvků vypadá na různých velikostech displeje stejně. Obrazovka tabulatury totiž obsahuje prvky, jejichž přesné zobrazení je závislé na poměrech svých velikostí. Na různých velikostech obrazovek by se rozvržení těchto prvků mohlo změnit.

Aplikace byla při vývoji testována na mobilním telefonu Xperia L s verzí API 17. Minimální podporovaná verze API je 16. V rámci uživatelských testů byla aplikace testována na verzích API 16, 21 a 25. Různé velikosti obrazovek byly testovány pomocí emulátorů.

Díky tomu, že byla aplikace vyvíjena na starší verzi API, proběhlo nasazení na jiné verze bez větších problémů. Největší problémy dělala verze 5.0 (API 21), kde bylo potřeba přepsat některá volání funkcí. Různé velikosti obrazovek aplikaci nečinily problém.

### 7.2 Testování uživatelského rozhraní

Uživatelské rozhraní bylo testováno přímo budoucími uživateli – tedy členy skupiny Insa-neTV. Nutno podotknout, že všichni účastníci testování měli základní znalosti hudební teorie a všichni měli už z minulosti psaní tabulatury osvojené z jiného editoru.

V rámci prvního testu byla uživatelům zadána série úkolů při jejichž plnění byli pozorováni. Cílem bylo zjistit jak se noví uživatelé orientují v uživatelském rozhraní a jak dlouho jim trvá vykonání jednotlivých úkonů. Mezi zadanými úkoly například bylo:

1. Přidej novou píseň s názvem *Skákal pes* a názvem interpreta *Myslivec*. Nastav délku na 4:20 a tempo na 220 bpm.
2. Změň přidané písni název.
3. Vytvoř nový seznam skladeb a přiřaď do něj skladbu.
4. Vytvoř další skladbu a přidej ji do seznamu skladeb z předchozího úkolu.

5. Změň pořadí skladeb v seznamu skladeb.
6. Přidej sekci *Refrén* do struktury některé skladby.
7. Změň pořadí sekcí struktury.
8. V sekci *Refrén* vytvoř poznámku s názvem *Note* a textem *Lalala*.
9. Přidej nový nástroj *Bass* do tabulatury a následně mu změň ladění.
10. Přepiš tabulaturu podle vzoru do aplikace.

Pro další fázi testování jsem do své verze aplikace připravil materiály pro další kapelní zkoušku. Cílem bylo zjistit, jak se zrychlí přípravná fáze zkoušky pokud budou všechny potřebné materiály v aplikaci. Poslední test představoval používání aplikace při hraní na hudební nástroj. Uživatelé v tomto testu interpretovali skladbu podle zaznamenané tabulatury. Cílem bylo zjistit, zda je forma zapsané tabulatury dostatečně přehledná.

### 7.3 Vyhodnocení výsledků

První fáze testování odhalila několik nedostatků v návrhu i v implementaci. Přidání a editace základních položek uživatelům nedělala problémy. Problém nastal až při přidávání sekcí struktury, kdy uživatele mátl plovoucí akční tlačítko, které má funkci vytvoření nové poznámky. Ve většině případů jej stiskli v domnění, že jím přidají novou sekci. Stížnost byla také na proces přidání poznámky, kde se uživatelům nelíbilo, že je možnost prvotně vytvořit pouze prázdnou poznámku a text dopsat až po přidání. Naopak chválili přehlednost jejich zobrazení.

Pro změnu ladění nástroje uživatelé vybírali symboly na začátku tabulatury, kde bylo aktuální ladění zobrazené. Patříčnou funkci v kontextovém menu našli až po dalším zkoumání. Ovládání zápisu tabulatury si uživatelé rychle osvojili a chválili jeho intuitivnost. Jediná stížnost byla na změnu délky již zapsaného symbolu, kdy se musí vložit znovu a nestačí jen vybrat novou délku.

Přípravná fáze zkoušky se při použití materiálů z aplikace zkrátila téměř o 20 minut. Tuto hodnotu je však třeba brát pouze orientačně. Zda má aplikace skutečně vliv na zrychlení chodu přípravné fáze zkoušek bude možné pozorovat až po delším používání. Interpretace hudby na základě tabulatury v aplikaci nečinila problém. Ukázalo se však, že když se uživatel věnuje hře na hudební nástroj a s aplikací nijak neinteraguje, tak zhasne podsvícení obrazovky. Uživatel pak musí přerušit hru na nástroj a obrazovku rozsvítit. Bylo by tedy dobré dát uživateli možnost zvolit, jestli obrazovku při prohlížení tabulatury po nějaké době automaticky vypínat nebo ne.

Další fáze testování proběhne v rámci publikace aplikace v internetovém obchodě Google Play, kde bude nejprve zveřejněna alfa verze aplikace, kdy je dostupná pouze vybraným testerům. Poté se aplikace přepne do beta verze, kdy jí může testovat široká veřejnost. V této fázi uživatelé mohou poskytnout zpětnou vazbu o tom, jaké funkce jim v aplikaci chybí apod. Po odladění aplikace bude veřejně dostupná stabilní verze.

## Kapitola 8

# Závěr

Cílem práce bylo navrhnout a implementovat aplikaci pro zjednodušení správy tvorby hudební skupiny. Stěžejním úkolem pak byl návrh a implementace editoru pro zápis hudby ve formě tabulatury. Před samotným návrhem bylo nutné sepsat požadavky na aplikaci na základě podnětů a pozorování skupiny InsaneTV. Poté byla provedena analýza již existujících řešení, kde jsem se zaměřil na splnění vytyčených požadavků. Díky tomu jsem objevil mezeru v trhu, kterou jsem se rozhodl vyplnit. Při návrhu uživatelského rozhraní bylo potřeba brát v potaz obecné praktiky dobrého návrhu a vycházet z již zažitých zvyklostí. Aplikace pak byla implementována v jazyku Java pro mobilní zařízení fungující pod operačním systémem Android.

Výsledná aplikace je schopna spravovat skladby kapely, setřídít je do seznamů a ty pak přiřazovat ke koncertům. U těch je pak možné zobrazit informace jako datum, čas a místo hraní, seznam skladeb nebo popis. U každé písně je možné zobrazit její text, jednotlivé sekce struktury, ke kterým je možné zapisovat poznámky. K dispozici je také editor, kterým je možné zapsat hudební nápady nebo celé písně ve formě tabulatury. MyBandSong umožňuje hudebním skupinám jednoduše spravovat svou tvorbu a koncerty z jednoho místa, díky čemuž je možné ušetřit čas jak na kapelních zkouškách, tak při přípravě na koncerty.

Další vývoj aplikace bude směřovat k vylepšení současných konceptů. Jedním z nápadů je provázání sekcí struktury s tabulaturou. Díky tomu by se mohla struktura skladby spravovat přímo v aktivitě tabulatury. Editor tabulatury by bylo možné rozšířit o další pokročilé funkce, které by zpříjemnily a ulehčily zápis hudby. Mezi ně patří například *copy&paste* operace, další typy hudebních symbolů nebo intuitivnější přidávání taktů. V neposlední řadě pak také podpora dalších hudebních nástrojů a zlepšení výkonu aplikace.

V rámci zveřejnění alfa verze aplikace v internetovém obchodě Google Play pak budou zpracovávány další uživatelské podněty. Před vydáním stabilní verze mám v plánu vytvořit vhodné propagační materiály včetně webové prezentace, aby se aplikace dostala do širšího povědomí uživatelů.

# Literatura

- [1] Activities. *Android Developers*. [2015], [Online; navštíveno 12.11.2016].  
URL <https://developer.android.com/training/basics/activity-lifecycle/index.html>
- [2] Android Interfaces and Architecture. *Android Open Source Project*. 2016, [Online; navštíveno 21.11.2016].  
URL <https://source.android.com/devices/>
- [3] Android Studio. *Android Developers*. 2016, [Online; navštíveno 15.11.2016].  
URL <https://developer.android.com/studio/intro/index.html>
- [4] Dashboards. *Android Developers*. 2016, [Online; navštíveno 15.11.2016].  
URL <https://developer.android.com/about/dashboards/index.html>
- [5] Introduction to Android. *Android Developers*. 2016, [Online; navštíveno 12.11.2016].  
URL <https://developer.android.com/guide/index.html>
- [6] Material Design for Android. *Android Developers*. 2016, [Online; navštíveno 15.11.2016].  
URL <https://developer.android.com/design/material/index.html>
- [7] *Statista*. 2017, [Online; navštíveno 21.1.2017].  
URL <https://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/>
- [8] Gerou, T.; Lusk, L.: *Essential dictionary of music notation: the most practical and concise source for music notation*. Los Angeles, CA: Alfred Pub, 1996, ISBN 0-88284-768-6.
- [9] Hlavík, J.: 3. díl – Android programování – Životní cyklus a nový projekt. *ITnetwork.cz*. [2015], [Online; navštíveno 12.11.2016].  
URL <http://www.itnetwork.cz/java/android/tutorial-programovani-pro-android-v-jave-zivotni-cyklus-a-novy-projekt>
- [10] Johnson, J.: *Designing with the mind in mind: simple guide to understanding user interface design rules*. Amsterdam: Morgan Kaufmann, 2010, ISBN 978-0-12-375030-3.
- [11] Swain, J. P.: *Historical dictionary of baroque music*. Lanham, Md.: Scarecrow Press, 2013, ISBN 978-0810878259.

- [12] Čapka, D.: MVC architektura. *ITnetwork*. 2016, [Online; navštíveno 7.5.2017].  
URL  
<https://www.itnetwork.cz/navrhove-vzory/mvc-architektura-navrhovy-vzor/>
- [13] Šimůnek, M.: *Technika hry na elektrickou kytaru*. Praha 4, AUDIO PARTNER s.r.o., 2014, ISBN 978-80-905786-0-9.

# Přílohy

# Příloha A

## Obsah CD

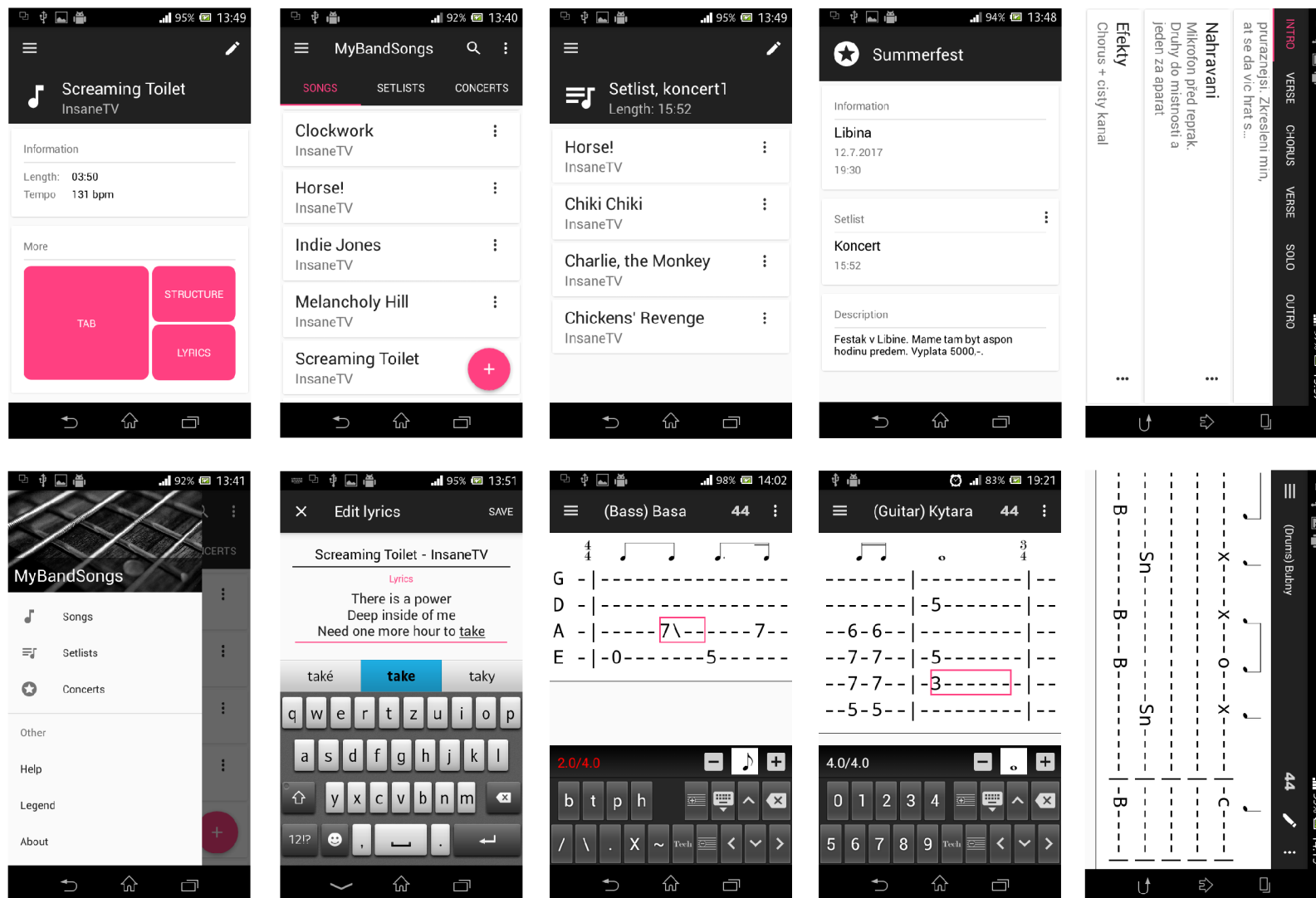
CD přiložené k této bakalářské práci obsahuje:

- text bakalářské práce ve formátu PDF,
- zdrojové soubory textu bakalářské práce vysázené v systému L<sup>A</sup>T<sub>E</sub>X,
- dokument ve formátu PDF obsahující manuál k aplikaci MyBandSong,
- zdrojové soubory aplikace ve formátu projektu Android Studio,
- instalační soubor ve formátu apk,
- textový soubor popisující obsah CD



**Příloha B**

**Snímky obrazovek aplikace**



Obrázek B.1: Snímky hlavních obrazovek aplikace. Odshora zleva: detail skladby, hlavní obrazovka se seznamem všech skladeb, detail seznamu skladeb, detail koncertu, struktura skladby horizontálně, skrývací menu, editace textu skladby, editace tabulatury baskytary, editace tabulatury kytary, prohlížení tabulatury pro bicí.