



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**MOBILNÍ APLIKACE PRO PODPORU VÝUKY HRY NA
KLÁVESOVÉ NÁSTROJE**

MOBILE APPLICATION FOR LEARNING TO PLAY KEYBOARD INSTRUMENTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ DOBROTKA

VEDOUcí PRÁCE

SUPERVISOR

Ing. MARTIN HRUBÝ, Ph.D.

BRNO 2019

Zadání bakalářské práce



22143

Student: **Dobrotka Tomáš**
Program: Informační technologie
Název: **Mobilní aplikace pro podporu výuky hry na klávesové nástroje**
Mobile Application for Learning to Play Keyboard Instruments
Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte programování aplikací pro iOS. Prostudujte metody výuky hry na klavír.
2. Navrhněte aplikaci pro mobilní zařízení s iOS, která bude uživatele provázet ve cvičeních hry na klavír. Aplikace bude dále umožňovat uživatelskou tvorbu lekcí hry a sdílení vytvořených lekcí mezi komunitou uživatelů.
3. Aplikaci implementujte.
4. Testujte výukový a editační mód aplikace se zapojením několika uživatelů.

Literatura:

- Keur, Ch., Hillegass, A.: iOS Programming: The Big Nerd Ranch Guide, Big Nerd Ranch Guides; 6 edition (January 6, 2017), ISBN-13: 978-0134682334

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Hrubý Martin, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 15. května 2019

Datum schválení: 1. listopadu 2018

Abstrakt

Cielom tejto bakalárskej práce je navrhnuť a implementovať aplikáciu pre mobilné zariadenia s operačným systémom iOS. Aplikácia slúži na podporu výučby hry na klávesové nástroje. Aplikácia zefektívňuje výučbu hry na klávesové nástroje. Ďalej aplikácia umožňuje vytvárať lekcie hry na klávesové nástroje s možnosťou zdieľania týchto lekcií medzi komunitou užívateľov. Pre úspešný vývoj bol najskôr vykonaný prieskum cieľovej skupiny, analýza požiadavkov, analýza správnych foriem vyučby hry na klávesové nástroje a následne návrh a implementácia. Výsledkom je funkčná mobilná aplikácia, ktorá je otestovaná a pripravená do reálnej prevádzky.

Abstract

The aim of the bachelor thesis is to design and implement an application for mobile devices running an operating system called iOS. The application is made for keyboard musical instruments teaching support. It also make the keyboard instruments teaching more effective. Moreover, the application is dispose of keyboard instruments lessons creating feature with the possibility of the lessons sharing between the user community. Firstly, there was done a target group survey, requirements analysis and an analysis of the correct ways how to teach keyboard musical instruments in order to successfully implement the application. After all the analyses, the application was designed respecting the analyses findings and finally, it was implemented. The result of the thesis consists of the functional mobile application which is tested and ready for an operation.

Klíčová slova

mobilná aplikácia, iOS, Swift, Xcode, Apple, CloudKit, CoreData, piano, MIDI, SoundFont, sf2

Keywords

mobile application, iOS, Swift, Xcode, Apple, CloudKit, CoreData, piano, MIDI, SoundFont, sf2

Citace

DOBROTKA, Tomáš. *Mobilní aplikace pro podporu výuky hry na klávesové nástroje*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Hrubý, Ph.D.

Mobilní aplikace pro podporu výuky hry na klávesové nástroje

Prohlášení

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Martina Hrubého, Ph.D. Uviedol som všetky literárne zdroje a publikácie, z ktorých som čerpal.

.....
Tomáš Dobrotka
14. května 2019

Poděkování

Ďakujem za odbornú pomoc vedúcemu bakalárskej práce pánovi Ing. Martinovi Hrubému, Ph.D., za jeho cenné rady a konzultácie pri písaní tejto práce.

Obsah

1	Úvod	2
2	Východiská práce	3
2.1	Programovanie mobilných aplikácií na platforme iOS	3
2.2	Výuka hry na klávesové nástroje	7
2.3	Štandard MIDI	10
2.4	Štandard SoundFont	12
2.5	Cloudové úložisko iCloud	13
2.6	Core Data	14
3	Návrh výslednej aplikácie	16
3.1	Prieskum cieľovej skupiny	16
3.2	Vytýčenie cieľov	16
3.3	Návrh užívateľského rozhrania	17
3.4	Návrh komunikácie mobilnej aplikácie	23
3.5	Návrh logiky aplikácie, ako aplikácia preberá systém metodiky výuky . . .	27
3.6	Návrh komunitného zdieľania lekcií	27
4	Implementácia mobilnej aplikácie pre podporu výuky hry na klávesové nástroje	29
4.1	Implementácia užívateľského rozhrania	29
4.2	Využitie MIDI súborov a import a export skladieb	30
4.3	Generovanie zvuku	32
5	Testovanie	34
5.1	Vstupné okolnosti testovania	34
5.2	Analýza dotazníku	35
5.3	Diskusia s užívateľmi	36
5.4	Odhalené nedostatky	37
5.5	Zhrnutie testovania	37
6	Záver	38
	Literatura	40

Kapitola 1

Úvod

Výučba hry na klávesové nástroje nie je vcelku triviálny proces, a je dosť náročné pobrať a pochytiť správne návyky pri výučbe. Preto je pochopiteľný fakt, že najlepšou formou učenia sa na klávesové nástroje je forma učenia sa s učiteľom, ktorý sprevádza žiaka počas jednotlivých lekcí a pri prípadnom zanedbaní nejakého aspektu správnej formy výučby, môže učiteľ žiaka v zápätí upozorniť na chybu. Je jasné, že nie je možné nahradiť v plnej forme učiteľa softwáreom.

Využitie softwaru avšak môže v začiatkoch výučby žiakovi pomôcť, a to tak, že šetrí čas učiteľovi, žiak nie je viazaný na učiteľa a teda proces výučby môže prebiehať kedykoľvek, pretože virtuálny učiteľ by mal byť dostupný kedykoľvek. A toto je práve dôvod, prečo by mal vzniknúť tento projekt. Umožniť žiakovi výučbu v čase, keď má voľno, náladu a chuť žiak. Keď k tomu pridáme fakt, že aplikácia je dostupná na mobilnom telefóne, tak sa z toho stáva hodnotný učiaci systém. Nakoľko v dnešnej dobe disponuje inteligentným telefónom drvivá väčšina obyvateľstva, je to teda ideálny spôsob, ako sprostredkovať túto službu čo najväčšiemu počtu ľudí. Ďalším cieľom aplikácie je aj užívateľská tvorba lekcí, kde by užívatelia mohli vytvárať vlastné výučbové lekcie, či už vo forme rôznych cvičení, alebo piesní, ktoré by boli automaticky dostupné všetkým užívateľom aplikácie. Každý užívateľ by mal mať možnosť zdieľať na sociálnych sieťach jednotlivé lekcie a podeliť sa so svojimi priateľmi o úspechoch alebo neúspechoch pri výučbe. Následne by mali mať priatelia možnosť jednotlivé lekcie vo forme príspevkov komentovať alebo likeovať.

Aby aplikácia spĺňala základné požiadavky na hodnotnú výučbu klávesových nástrojov, je potrebné, aby bola implementovaná funkcionálna jednotlivých aspektov dobrej formy výučby. Medzi tieto aspekty patrí výučba stupníc, výučba rôznych žánrov formou skladieb, zmena tempa jednotlivých skladieb, možnosť výučby hry každej ruky zvlášť, ale aj hry oboch rúk naraz, možnosť výberu štvorročných skladieb (skladby stavané pre hru dvoch ľudí na jednom klávesovom nástroji), možnosť cyklického opakovania jednotlivých pasáží v piesni.

Pri vypracovaní tohto projektu som sa snažil brať do úvahy všetky vyššie spomenuté aspekty.

Kapitola 2

Východiská práce

2.1 Programovanie mobilných aplikácií na platforme iOS

Na vývoj iOS aplikácii je ideálne použiť počítač so systémom OS X s požadovanými vývojarskými nástrojmi. Testovanie vytvorených aplikácii je možné skúšať na simulátore alebo na reálnom zariadení. Testovanie aplikácie na simulátore je zdarma a je súčasťou vývojarských nástrojov. Pokiaľ však chce programátor testovať vytvorenú aplikáciu na reálnom zariadení, je potrebné byť registrovaným vývojárom. Ročná cena tohoto členstva je \$99. V prípade podnikového riešenia je cena \$299, kde je poskytnutá možnosť distribúcie aplikácie mimo AppStore a teda možnosť distribúcie na zariadenia vnútri firmy. Posledná možnosť je určená univerzitám. V tomto prípade je možné aplikáciu testovať na reálnom zariadení, no nie je umožnené publikovať aplikáciu na AppStore.

Vývojové prostredie

Vývojovým prostredím pre tvorbu aplikácií pre platformu iOS je Xcode.

Pri vývoji v prostredí Xcode od Applu je potreba mať počítač s operačným systémom macOS. Pre vývoj bude stačiť ktorýkoľvek z momentálne uvedených typov v predaji. Dokonca to nemusí byť ani najnovší model. Nároky vývojového prostredia na výpočetný výkon a pamäťovú kapacitu sú pomerne rozumné, takže postačí aj lacnejší model.

Spoločne s aplikáciou Xcode sa stiahnu a nainštalujú všetky potrebné frameworky, nástroje pre debuggovanie, testovanie, pre správu certifikátov či odoslanie aplikácie do iTunes, ako aj nástroje spolu so simulátormi zariadení s iOS, v ktorých je možné spúšťať a ladit aplikácie priamo na vývojárskom počítači bez toho, aby sa museli zakaždým prenášať a spúšťať na skutočnom zariadení [16].

Vývojové prostredie Xcode je v obchode s aplikáciami pre macOS k dispozícii zadarmo (v minulosti bolo zadarmo len pre registrovaných vývojárov, ostatní si ho mohli kúpiť za symbolickú cenu 5 dolárov), je to bystrý ťah od spoločnosti Apple, ktorým chce získať záujem začínajúcich a hobby vývojárov. Pokiaľ má vývojár počítač s macOS, nemusí už do začiatku vývoja aplikácie nič investovať. A až po vytvorení prvej životaschopnej aplikácie, o ktorej si bude myslieť, že by o ňu mohol byť záujem, a po jej odladení v simulátore si môže zaplatiť ročnú vývojársku licenciu a začať aplikáciu šíriť cez App Store [16].

Instruments

Instruments pomáha k analýze chovania aplikácií k optimalizácií využitia zdrojov. Je to nástroj pre analýzu a testovanie výkonu, ktorý je súčasťou nástrojov v Xcode. Na rozdiel od iných nástrojov pre odhaľovanie chýb umožňuje zhromažďovať veľmi rozmanité typy dát a zobrazovať ich vedľa seba. To uľahčuje nájdenie zdroja problému napr. pri nadmernom využití pamäte [1].

Návrh a tvorba užívateľského rozhrania

Na Obrázku 2.1 je zobrazený nástroj Interface Builder, ktorý slúži na tvorbu užívateľského rozhrania. V nástroji Interface Builder je možné vytvoriť vzhľad jednotlivých komponentov ale aj celých obrazoviek. Vývojár tu nemusí napísať ani riadok kódu. Výhodou vytvárania vzhľadu v Interface Builderi je, že vzhľad obrazovky je vidieť okamžite, bez nutnosti spúšťania aplikácie narozdiel od vytváraníu vzhľadu pomocou kódu v aplikácií kde vzhľad je vidno až po spustení. Obrazovky aplikácie je možné vytvárať zvlášť v súboroch „xib“. Ďalej je možné vytvorenie „storyboardu“, ktorý tak ako xib, slúži k návrhu obrazovky aplikácie. Pri práci vo väčších tímoch sa však jeho používanie môže stať problematické, pokiaľ nie sú jednotlivé prípady použitia rozumne rozdelené do viacerých storyboardov. Ak by na jednom súbore malo pracovať viac ľudí, je skoro nemožné zlučovať zmeny.

Jednotlivé prvky sa vyberajú z ponuky „Library“ v pravom hornom rohu (Obrázok 2.1 - červený obdĺžnik). Celá tvorba prebieha štýlom „drag and drop“, čiže ťahaním komponentov na konkrétne miesto, na ktorom sa potom zväčší alebo zmenší podľa potreby. Pridať možno aj vlastné komponenty. Najprv je potreba napozícovať základnú komponentu UIView. Pri tej potom treba nastaviť konkrétnu triedu, ktorá má byť pre prvok použitá. Ak chceme mať vlastnú komponentu vykreslenú v Interface Builder tak, ako by vyzerala v aplikácii, je potrebné ju deklarovať s direktívou @IBDesignable. Je možné nastaviť aj parametre jednotlivých prvkov, nech už sa jedná o farbu, písmo, veľkosť alebo farbu pozadia prostredníctvom užívateľského rozhrania [17].

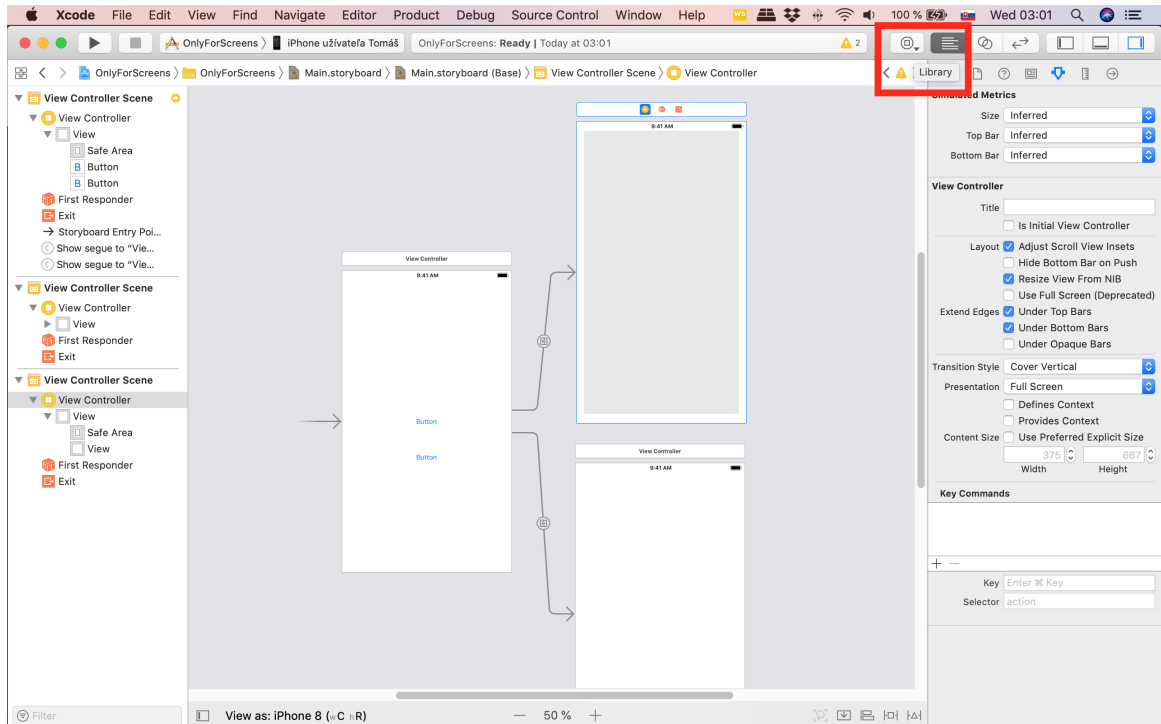
Pomocou tzv. Segues je možné definovať akým spôsobom sa má zobrazíť nová obrazovka. Segues sa dajú nastaviť v storyboardoch a slúžia teda na zobrazenie ďalších obrazoviek. Ku každému Segue možno nastaviť identifikátor. Identifikátor slúži na identifikáciu pri vyvolaní danej konkrétnej Segue. Vďaka tomu možno nastaviť ďalšie akcie, ktoré sa majú vykonať pred zobrazením novej obrazovky [17].

Autolayout

Autolayout [5] dynamicky počíta veľkosť a pozíciu všetkých prvkov obrazovky podľa obmedzení, ktoré majú nastavené. Pri zmene veľkosti obrazovky sa veľkosť aj pozícia prepočítajú tak, aby platili všetky podmienky obmedzenia. Týmto spôsobom sa dá tvoriť užívateľské rozhranie, ktoré je schopné sa prispôbiť interným a externým zmenám.

Interné zmeny

- Zmena zobrazovaného obsahu
- Aplikácia podporuje dynamic types



Obrázek 2.1: Interface Builder so storyboardom.

Externé zmeny

- Podpora rôznych veľkostí displejov
- Rotácia zariadenia
- Zmena Split View na zariadení iPad
- Zobrazenie indikátorov aktívnych hovorov a zvukov
- podpora size classes
- Zmena veľkosti status baru

Obmedzenia zvyčajne predstavujú vzťah medzi dvoma prvkami. Automatické rozloženie potom vypočíta veľkosť a umiestnenie každého zobrazenia na základe týchto obmedzení. To vytvára rozloženie, ktoré dynamicky reaguje na interné aj externé zmeny [5].

Programovací jazyk Swift

Moja aplikácia je vytvorená v programovacom jazyku Swift. V nasledujúcom texte je tento jazyk v krátkosti popísaný.

Ako sa píše v knihe pána E. Lacka [16], že ešte pred rokom 2014 sa aplikácie pre mobilné zariadenia využívajúce operačný systém iOS programovali vo vývojarskom prostredí Xcode v programovacom jazyku Objective-C. Je to jedna z objektovo orientovaných variant programovacieho jazyka C. Pokrok však ide dopredu, a modernizácii sa nevyhnú ani programovacie jazyky. Cieľom modernizácie je zjednodušený a zrýchlený vývoj a prehľadnejší

kód, v ktorom programátori už na prvý pohľad odhalia niektoré chyby. Medzi najväčšie výhody programovacieho jazyka Swift patria lepšia čitateľnosť kódu, rýchlosť a redukcia dĺžky kódu.

Programovací jazyk Swift je pomerne mladý, prvý krát bol prezentovaný v júni roku 2014 na konferencii Apple WWDC (Worldwide Developers Conference). V roku 2014 bola k dispozícii aj prvá verzia. Táto verzia bola integrovaná do vývojového prostredia Xcode vo verzii 6. O rok neskôr Apple zverejnil zdrojový kód Swiftu, od tej doby je tento programovací jazyk typu open-source. Zdrojový kód Swiftu je dostupný vo verejnom repozitári firmy Apple na Githubu.

Swift je striktne typovaný, rýchly a interaktívny programovací jazyk, ktorý kombinuje najlepšie moderné programátorské myslenie s osvedčenými postupmi, či už softwarových inžinierov Applu, alebo open-source komunity. Kompilátor je optimalizovaný na výkon, a jazyk je optimalizovaný na efektívny vývoj. Apple v dobe uvedenia jazyka Swift porovnal na svojich stránkach tento jazyk s inými jazykmi z hľadiska výkonu a tvrdil, že Swift je 2,6 krát rýchlejší ako Objective-C a 8,4 krát rýchlejší ako Python [16].

Apple umožňuje experimentovať s kódom v špeciálnom prostredí Playground, takže je možné sa naučiť jazyk bez toho, aby začiatočník musel riešiť aspekty projektu reálnej aplikácie. Prvú aplikáciu je možné začať vytvárať až po osvojení Swiftu na primeranej úrovni.

Kód vo Swiftu je kompilovaný a optimalizovaný tak, aby čo najviac využil moderný hardware mobilných zariadení Apple. Syntax a štandardné knižnice boli navrhnuté na základe osvedčeného princípu, aby spôsob zápisu kódu bol jasný a prehľadný. Kombinácia bezpečnosti a rýchlosti robí zo Swiftu ideálnu voľbu pre všetky typy aplikácií, od najjednoduchších až po náročné, ako sú práce s databázami, či emulátor iného operačného systému.

Tvorcovia Swiftu sa okrem jednoduchosti sústredili aj na bezpečnosť, tento programovací jazyk je teda z hľadiska bezpečnosti pomerne striktný. Vyžaduje napríklad inicializáciu premenných pred ich prvým použitím, definovanie hodnôt konštánt, či neumožňuje, aby objekty bez toho, aby to programátor pri inicializácii vyslovene pripustil, mali hodnotu nil. Pokiaľ to pre nejaký objekt vedome pripustí, musí takúto situáciu v kóde následne ošetriť [16].

Spätná kompatibilita s Objective-C

Pretože sa do polovice roku 2014 aplikácie vyvíjali hlavne v jazyku Objective-C, je vo vývojovom prostredí Xcode zabudovaná kompatibilita s Objective-C. Znamená to, že v nových projektoch využívajúcich moderný Swift je možné využívať knižnice a frameworky vytvorené v Objective-C. Na ich integráciu je potrebné definovať hlavičkový súbor, takzvaný bridging header, ktorý umožní použitie kódu Objective-C v projektoch písaných vo Swiftu. Príchod Swiftu neznamená žiadne problémy s aplikáciami písanými v Objective-C, pretože v jednej aplikácii pre iOS môže existovať kód písaný v Objective-C aj kód písaný vo Swiftu. Takže nie je problém v písaní nových modulov v programovacom jazyku Swift, ktoré sú kompatibilné s existujúcim kódom v Objective-C [16].

2.2 Výuka hry na klávesové nástroje

Technika hry na klavíri

Moja bakalárska práca si nekladie za cieľ vytvárať metodiku hry na klavír, ale preberá metodiku publikovanú v knihe Chuan C. Changa s názvom *Fundamentals of Piano Practice* [13] a popis metodiky si dovoľujem v nasledujúcom texte interpretovať vlastnými slovami.

Aby sme pochopili význam slovného spojenia technika hry na klavíri, musíme pochopiť čo je to technika, pretože nepochopenie by mohlo viesť k nesprávnym interpretáciám. Správne pochopenie pomôže k lepšiemu pochopeniu cvičebných metód. Najbežnejšie nedorozumenia sú tie, že metóda je nejaké dedičstvo prstovej zručnosti. Nie je to však tak. Vrodená obratnosť uznávaných klavíristov a obyčajných ľudí nie je tak odlišná. To znamená, že prakticky každý sa môže naučiť hrať dobre na klavíri. Sú početné príklady ľudí s mentálnym postihnutím s obmedzenou koordináciou, ktorí vykazujú neuveriteľný hudobný talent (savant). Mnoho ľudí je oveľa ohybnejších, ale nevedia viesť hudobné pasáže kvôli nedostatku nejakej jednoduchej, ale úplne zásadnej informácii. Získanie techniky je väčšinou proces rozvoja mozgu/nervov, nie rozvoja sily v prstoch.

Technika je schopnosť vykonávať veľké množstvo rôznych klavírnych pasáží. Preto to nie je zručnosť, ale skĺbenie mnohých zručností.

Výhodou je, že klavírne zručnosti sa dajú naučiť v krátkom čase, v prípade, že sú aplikované správne postupy učenia.

Dve fázy získavania zručností:

- Zistenie ako sa majú pohybovať prsty.
- Podmienenie mozgu, svalov tak aby sa tieto pohyby vykonávali s ľahkosťou a kontrolou.

V skutočnosti sa pri učení hre na klavír zlepšuje mozog. Vlastne sa robí múdrejším a zlepšuje si svoju pamäť, čo je dôvod, prečo správne učenie klavíra má tak veľa výhod, ako je napríklad úspech v škole, schopnosť lepšie sa vyrovnáť s každodennými problémami, a schopnosť udržať pamäť dlhšie. To je dôvod, prečo zapamätávanie je neoddeliteľnou súčasťou získavania techniky.

Ďalšia mylná predstava o technike je, že akonáhle sa prsty stanú dostatočne zručnými, môže si žiak zahrať čokoľvek. Takmer každá rozdielna pasáž je nové dobrodružstvo, a musí sa naučiť znova. Skúsení klavíristi sa zdajú byť schopní hrať len tak akoby nič, pretože majú skúsenosti so všetkými vecami, na ktoré často narazia, a vedia, ako sa učiť nové veci veľmi rýchlo. Sú veľké triedy pasáží, ako napríklad stupnice, ktoré sa objavujú často. Znalosť toho, ako ich hrať pokryje významné časti väčšiny skladieb. Preto je cvičenie stupníc veľmi dôležité.

Technika, hudba, hranie v mysli

Ak sa sústredíme len na vývoj „prstovej techniky“ a zanedbávame hudbu počas tréningu, môžeme nadobudnúť nehudobné návyky. Nehudobné hranie je nežiadúce, a je to jedna z foriem chyby. Prejavuje sa keď je prítomné publikum, títo študenti robia podivné chyby, ktoré nerobili počas „cvičenia“. To sa deje preto, že študenti praktizovali bez ohľadu na hudbu, ale zrazu si uvedomili, že hudba musí byť teraz pridaná preto, že niekto počúva.

Študenti potrebujú cvičiť hudobne, aby získali techniku. Čo je správne pre uši a mozog, to sa ukazuje byť správne pre ľudský hrací mechanizmus. Je dôležité si skladbu vypočuť [13].

Základné postupy pri cvičení na klavíri:

- Vypočuť si predstavenie/lekciiu.

Najlepší spôsob ako začať proces učenia sa novej skladby.

- Cvičenie ťažkých úsekov ako prvé.

Najťažšie časti budú trvať najdlhší čas na naučenie, takže by im malo byť venované čo najviac času cvičenia. Pretože závery skladieb sú vo všeobecnosti najťažšie, bude sa žiak často učiť väčšinu skladieb od konca.

- Zvoliť si krátky úsek na cvičenie.

Tento trik má možno najväčší vplyv na zníženie času cvičenia z mnohých dôvodov.

Nácvik krátkych úsekov umožňuje cvičiť ich desiatky, dokonca aj stovky krát, počas niekoľkých minút. Použitie týchto rýchlych opakovaní je najrýchlejší spôsob, ako naučiť svoje ruky nové pohyby.

Cvičenie rúk, každú zvlášť: získavanie techniky

V podstate 100% rozvoja techniky je dosiahnuté tým, že sa každá ruka cvičí zvlášť. Netreba sa snažiť vypracovať si techniku prstov/ruky hraním rukami spolu, pretože je to oveľa ťažšie, časovo náročné. Vhodný je výber dvoch krátkych pasáží, jednu pre pravú ruku a druhú pre ľavú. Je vhodné cvičiť jednu ruku až pokiaľ sa nezačne unavovať, potom je potreba prepnúť na druhú ruku. Prepína sa medzi rukami pred tým, ako sa cvičiacia ruka stáva unavenou.

Je potrebné zdôrazniť, že cvičenie s rukami zvlášť je len pre ťažké pasáže, ktoré žiak nedokáže zahráť. Ak je žiak schopný hrať s rukami spolu, nie je v tom prípade vhodné cvičiť ruky zvlášť [13].

Rýchlosť, voľba rýchlosti pri cvičení

Pán Chuan C. Chang vo svojej knihe[13] taktiež spomína dôležitosť tempa pri cvičení. Dôležité je dostať sa do rýchlosti tak rýchlo, ako je to možné.

Ak žiak chce meniť rýchlosť, najskôr sa treba dostať na nejakú zvládnuteľnú „maximálnu rýchlosť“, pri ktorej môže zahráť presne. Potom by mala nasledovať rýchlejšia hra a žiak si určite všimne ako treba zmeniť pohyb prstov. Potom treba použiť tento pohyb a hrať na predchádzajúcej „maximálnej rýchlosti presného hrania“. To by teraz malo byť výrazne jednoduchšie. Následne žiak cvičí chvíľu na tejto rýchlosti, a potom skúsi pomalšiu rýchlosť, aby sa ubezpečil, že je úplne uvoľnený a úplne presný. Potom by mal celý postup opakovať. Týmto spôsobom môže zvyšovať rýchlosť v zvládnuteľných skokoch a pracovať na každej potrebnej zručnosti zvlášť. Vo väčšine prípadov by malo byť možné hrať väčšinu nových skladieb, aspoň v malých úsekoch s rukami zvlášť, na konečnej rýchlosti v priebehu prvého cvičenia lekcii. Na začiatku sa môže dostať do rýchlosti pri prvom cvičení zdať nedosiahnuteľné, ale s cvičením môže každý študent dosiahnuť tento cieľ prekvapivo rýchlo.

Dôležitosť pomalého hrania

Vždy treba ukončiť sedenie tým, že sa lekcia zahrá pomaly aspoň raz. Mal by sa tiež pestovať zvyk robiť to pri prepínaní rúk počas tréningu rúk zvlášť, pred prepínaním hrať pomaly aspoň raz. To môže byť jedným z najdôležitejších pravidiel výučby hry na klavír, pretože má tak neúmerne veľký vplyv na zlepšenie techniky. Je to výhodné ako aj pre okamžité tak aj pre dlhodobé zlepšenie. Jedným z dôvodov, prečo to funguje môže byť, že sa žiak môže úplne uvoľniť. Ďalším dôvodom môže byť to, že ma tendenciu nabráť viac zlých návykov, než si uvedomuje pri hraní rýchlo a môže „zmazať“ tieto návyky pomalým hraním. Na rozdiel od intuície, pomalé hranie bez chýb je ťažké (pokiaľ predtým úplne nezvládol pasáž). Takže pomalá hra je dobrý spôsob, ako otestovať, či sa žiak túto skladbu skutočne naučil.

Dôležitá zručnosť cvičiť pri hraní pomaly je myslieť dopredu za práve hranú hudbu. Pri cvičení novej skladby rýchlo je tendencia byť mentálne pozadu s hudbou, a to sa môže stať zvykom. To je zlé, pretože tým žiak stráca kontrolu. Žiak by mal teda myslieť dopredu pri hraní pomaly a pokúsiť sa udržať tento náskok, keď sa dostane späť do tempa. Myslením dopredu môže obvykle predvídať nepodarky alebo prichádzajúce ťažkosti a ma čas, aby prijal vhodné opatrenia.

Slabá ľavá ruka, jedna ruka učí druhú

Študenti, ktorí necvičia ruky zvlášť budú mať vždy silnejšiu pravú ruku ako ľavú ruku. To sa stane preto, že pasáže pre pravú ruku sú vo všeobecnosti ťažšie, technicky. Ľavá ruka má tendenciu dostať pasáže, ktoré vyžadujú väčšiu silu, ale často zaostávajú v rýchlosti a technike. „Slabšia“ ruka tu znamená technicky slabšia, nie silovo. Metóda rúk zvlášť zrovnováži ruky, pretože automaticky dá slabšej ruke viac práce. V pasážach, ktoré jedna ruka môže hrať lepšie ako druhá, je táto lepšia ruka často najlepší učiteľ. Ak žiak chce, aby jedna ruka učila druhú, vyberie si krátky úsek a zahrá ho rýchlo s lepšou rukou, potom okamžite zopakuje slabšou rukou, jednu oktávu od seba, aby sa zabránilo kolíziám. Žiak zistí, že slabšia ruka sa často „chytí“ alebo „získa predstavu“ o tom, ako to robí lepšia ruka. Prstoklad by mal byť podobný, ale nie je potrebné aby bol zhodný. Akonáhle slabšia ruka „dostane nápad“, postupne ju odstavuje tým, že hrá slabšiu ruku dvakrát než silnejšiu, potom trikrát voči jednej, atď. Táto schopnosť učiť jednou rukou druhú je dôležitejšia, než si väčšina ľudí uvedomuje. Vyššie uvedený príklad riešenia jedného konkrétneho technického problému je len jeden príklad - čo je dôležitejšie, tento koncept sa týka prakticky každého cvičebného sedenia. Základným zmyslom tejto širokej použiteľnosti je, že jedna ruka hrá vždy niečo lepšie ako druhá, či je to relaxácia, rýchlosť, pokojné ruky, a nespočetné množstvo pohybov prstov/ruky (palec cez, ploché prsty, a pod.) - hocičo nové, čo sa žiak snaží naučiť. Preto, akonáhle sa naučí tento princíp pomocou jednej ruky učiť druhú, bude ho používať stále [13].

Cyklické opakovanie

Cyklické opakovanie je najlepší spôsob budovania techniky pre veci, ako sú nové alebo rýchle pasáže, ktoré žiak nemôže zvládnuť. Opakovanie v cykloch (tiež nazývané „slučky“) znamená zobrať úsek a hrať ho opakovane. Zvyčajne nepretržite, bez prestávky.

Vykonávanie cyklov je v podstate čisté opakovanie, ale je dôležité ho používať takmer ako procedúru proti opakovaniu, spôsob ako sa vyhnúť bezduchému opakovaniu. Myšlienka za vykonávaním cyklov je, že žiak získa techniku tak rýchlo, že sa eliminujú zbytočné, bezduché

opakovania. Aby nedošlo k získaniu zlozvykov, žiak by mal meniť rýchlosť a experimentovať s rôznymi pozíciami ruky/paže/prstov pre optimálne hranie a vždy pracuje pre relaxáciu. Nemal by sa snažiť cyklíť presne to isté príliš mnoho krát. Mal by hrať ticho (aj úseky ktoré majú byť hrané nahlas), kým nedosiahne techniku, získava rýchlosť najmenej o 20% vyššiu nad konečnou rýchlosťou, a ak je to možné, až dvojnásobnú než je konečná rýchlosť. Viac ako 90% času kedy žiak vykonáva cykly by malo byť pri rýchlostiach, ktoré môže zvládnuť pohodlne a presne. Potom vykonáva cykly postupne až na veľmi pomalé rýchlosti. Žiak je s tým hotový, ak úsek môže zahrať ľubovoľnou rýchlosťou po dlhšiu dobu, bez toho, aby sa pozeral na klávesy, úplne uvoľnene a s plnou kontrolou. Možno zistí, že niektoré stredné rýchlosti mu spôsobujú ťažkosti. Mal by cvičiť tieto rýchlosti, pretože môžu byť potrebné, keď začne cvičiť ruky spolu.

Ak technika vyžaduje opakovanie 10000 cyklov (typická požiadavka naozaj náročného materiálu), vykonávanie cyklov žiakovi umožní mať ich hotové v najkratšom možnom čase. Reprezentatívny čas vykonania cyklu je asi 1 sec, takže 10000 cyklov bude trvať kratšie ako 4 hodiny. Ak žiak vykonáva cyklus tohto úseku asi 10 min. denne, 5 dní v týždni, bude 10000 cyklov trvať skoro mesiac. Je jasné, že veľmi ťažký materiál bude trvať mesiace na naučenie s najlepšimi metódami, a oveľa dlhšie, ak bude používať menej účinné metódy.

Vykonávanie cyklov je procedúra spôsobujúca potenciálne najviac zranení z ktorejkoľvek cvičebnej procedúry, takže treba byť opatrný. Netreba to prehnať v prvý deň cvičenia, aby žiak zistil, čo sa stane ďalší deň. Ak nič nie je zranené, na druhý deň, môže pokračovať alebo zvýšiť tréning vykonávania cyklov. Predovšetkým, keď pracuje na cykloch, pracuje na dvoch naraz, jeden pre pravú ruku a ďalší pre ľavú ruku, takže môže často prepínať ruky. U mladých ľudí môže prehnané cyklenie spôsobiť bolesť, v tomto prípade treba zastaviť cykly, a ruka by sa mala zotaviť v priebehu niekoľkých dní [13].

2.3 Štandard MIDI

Moja aplikácia *The Piano Master* používa súbory vo formáte MIDI v ktorých sú uložené potrebné informácie pre reprezentáciu piesní. V nasledujúcom texte je popísaný štandard MIDI.

MIDI je štandardný hudobný technologický protokol, ktorý spája produkty z rôznych spoločností vrátane digitálnych hudobných nástrojov, počítačov, tabletov a smartfónov. MIDI sa používa každý deň na celom svete hudobníkmi, DJ-mi, hudobnými producentmi, učiteľmi v oblasti hudby, umelcami a fanúšikmi na vytváranie, predvádzanie, učenie a zdieľanie hudobných a umeleckých diel [8].

Zavedenie MIDI technológie prinieslo viaceré výhody. Jednou z veľkých výhod je značná redukcia kabeláže medzi jednotlivými zariadeniami a s tým spojená menšia náročnosť na rôzne ďalšie prevodníky, ktoré zaisťovali kompatibilitu medzi zariadeniami rôznych značiek. Ďalšou veľkou výhodou je univerzalita použitia.

Pomocou MIDI sa neprenášajú žiadne zvukové informácie, ale len správy o tom, ktorý nástroj alebo ktorá časť nástroja práve znie, aký tón tento nástroj vydáva, aké efekty boli použité pri interpretovaní zvuku a podobne. Každá takáto udalosť je vyhodnotená zariadením, ktoré je schopné MIDI informáciu interpretovať. Týmto spôsobom môžeme napríklad zaznamenať na počítači, kedy bol stlačený ktorý kláves elektrického klávesového nástroja, môžeme jej priradiť zvuk jedného z palety nástrojov a vytvoriť zvukový výstup. V MIDI má každá nota číselnú reprezentáciu podľa poradia na stupnici. Príkladom môže byť tón c3, ktorý má v MIDI štandarde hodnotu 60. MIDI rozhranie je sériové a je možné v MIDI obsiahnuť 16 kanálov.

MIDI všeobecne

MIDI ako termín je skratka - Musical Instruments Digital Interface, čo by sa dalo preložiť ako digitálne rozhranie pre hudobné nástroje. Prekvapivo samotný názov nie je scestný, vypovedá dosť o tom, čo to MIDI skutočne je. Ide o komunikačný protokol a potrebné hardwarové a softwarové vybavenie a štandardy, teda v podstate o celý svet, kde medzi sebou môžu komunikovať digitálne hudobné zariadenia, či už ide o sekvencery, syntezátory, mixážne pulty a iné. Rovnako ako v inej komunikácii, aj digitálne zariadenia potrebujú vedieť, ako sa majú určité „slová“ (dátá) vykladať, ako im majú porozumieť a čo majú na základe daného povelu urobiť. Preto bolo potrebné stanoviť normu, ktorou sa stala práve MIDI, a ktorá uzrela svetlo sveta už v roku 1982. Dodnes je táto technológia spravovaná organizáciou MIDI Manufacturers Association [15].

MIDI informácia

Existuje niekoľko typov informácií prenášaných cez MIDI rozhranie. Patria sem napríklad tzv. SysEx (System Exclusive) správy. Tieto správy slúžia predovšetkým k nastavovaniu pokročilých vlastností medzi zariadeniami, napr. požiadavky pre výpis textu na display, nastavenie celkovej hlasitosti atd. Do kategórie SysEx informácií spadajú aj špeciálne správy slúžiace k nastavovaniu konkrétneho parametru pri špecifických zariadeniach. Takéto správy nie sú štandardizované a väčšinou sú dedefinované výrobcom daného zariadenia [15].

Ďalšou skupinou MIDI informácií sú kanálové MIDI správy. Tieto informácie sú určené k posielaniu požiadaviek na ohýbanie nôt, prehratie noty, zastavenie prehrávania atd.

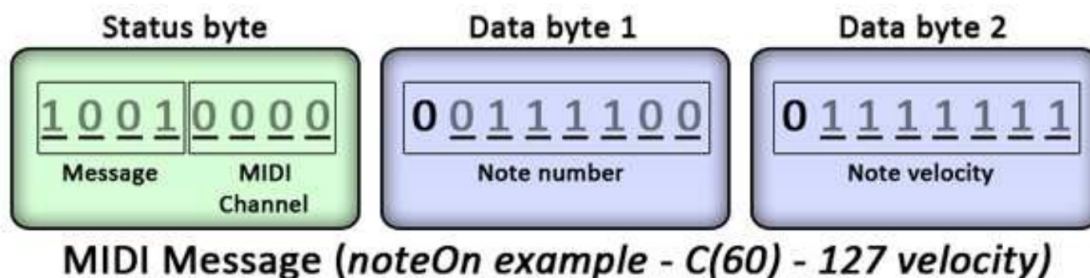
Poslednou skupinou MIDI informácií sú systémové správy (MIDI system messages). Tieto správy sú určené pre nastavovanie hodnôt, ktoré sú spoločné pre všetky kanály. Príkladom týchto správ je žiadosť o naladenie syntezátoru alebo voľba kanálu.

Štruktúra kanálovej MIDI informácie

Kanálová MIDI informácia má veľkosť 3 bajty. Na Obrázku 2.2 je uvedená štruktúra tejto informácie s príkladom správy s úlohou zahrať notu C s maximálnou hlasitosťou na prvom kanáli. V MIDI správach sa kanále číslujú od 0, binárne to je hodnota 0000.

V status bajte sa nachádzajú dve štvorbitové informácie. Prvá udáva typ požiadavky, druhá udáva kanál. Ďalej sa tam nachádzajú dva dátové bajty, ktoré nesú informácie o čísle noty - prvý bajt, a o jej sile - druhý bajt.

Podobnú schému majú aj systémové MIDI správy, líšia sa číslom požiadavky a významom hodnôt dátových bajtov [7].



Obrázek 2.2: Štruktúra kanálovej MIDI informácie s príkladom správy noteOn tónu C s maximálnou hlasitosťou na prvom kanáli.

2.4 Štandard SoundFont

Aplikácia *The Piano Master* vytvára zvukovú reprezentáciu nôt popísaných v MIDI súbore pomocou súboru vo formáte „sf2“. Táto kapitola popisuje formát „sf2“ ako aj obchodnú značku SoundFont.

Špecifikácia

SoundFont je obchodná značka vzťahujúca sa k súborovému formátu „sf2“ a taktiež k súvisiacej technológii vytvorenej pre vyplnenie medzery medzi nahrávaným a syntetizovaným zvukom. SoundFont obsahuje zvukové vzorky a informácie pre ich organizáciu a spracovanie. Podľa týchto informácií môže generátor tónov riadený MIDI dátami, generovať tóny požadovaných farieb, výšok, hlasitosti a ďalších parametrov [9].

Zvukové banky *sf2* sa dajú použiť v softwarových generátoroch tónov. Zvukové banky *sf2* zaberajú napr. v porovnaní s formátom .wav niekoľkonásobne menšiu veľkosť na disku.

Najnovšia verzia súborov vo formáte SoundFont je 2.04. Je založená na formáte RIFF. Podrobný popis tohto formátu je v špecifikácii, na stránkach spoločnosti [11].

SoundFont slúži ako alternatíva k syntezátorom, je to technológia, ktorá umožňuje použiť akýkoľvek zvuk ako nástroj MIDI a nezostaneme tak odkázaní iba na zvuky, ktoré je schopná vygenerovať naša zvuková karta.

Zvuky sú uložené v súboroch SoundFontu, ktoré majú príponu .sf2 (pokiaľ sa jedná o dnes najrozšírejší štandard SoundFont 2). Tieto súbory sa nahrajú do pamäte, odkiaľ sú prehrávané [12].

História

Pôvodný formát súboru SoundFont bol vyvinutý na začiatku 90. rokov v E-mu Systems a Creative Labs. Špecifikácia tejto verzie nebola nikdy zverejnená. Prvé a jediné zariadenie, ktoré využívalo túto verziu, bolo Sound Blaster AWE32 od Creative v roku 1994. Súbory v tomto formáte majú zvyčajne príponu .SBK alebo .SB2.

SoundFont 2.0 bol vyvinutý v roku 1996. Tento formát súboru zovšeobecnil reprezentáciu dát v SoundFont, predefinoval niektoré funkcie vrstvenia nástroja v rámci formátu, pridal podporu stereo vzoriek a odstránil niektoré neznáme negatívne vlastnosti prvej verzie, ktorej správanie bolo ťažké špecifikovať. Špecifikácia tejto verzie bola zverejnená ako verejná s cieľom vytvoriť z formátu SoundFont priemyselný štandard. Všetky zariadenia kompatibilné so systémom SoundFont 1.0 boli aktualizované tak, aby podporovali formát SoundFont 2.0 krátko po jeho zverejnení, a preto sa verzia 1.0 stala za krátky čas zastaranou. Súbory v tomto a všetkých ostatných formátoch 2.x majú zvyčajne príponu .sf2.

Verzia 2.01 (zvyčajne nesprávne nazvaná 2.1) formátu súboru SoundFont bola zavedená v roku 1998 s produktom zvukovej karty E-mu s názvom Audio Production Studio. Verzia 2.01 pridal funkcie umožňujúce zvukovým dizajnérom konfigurovať spôsob, akým MIDI kontroléry ovplyvňujú parametre syntezátora. Formát 2.01 je obojsmerne kompatibilný s 2.0, čo znamená, že syntezátory schopné vykresľovať formát 2.01 budú tiež schopné vykresľovať formát 2.0, a syntezátory, ktoré sú schopné vykresľovať iba formát 2.0, budú tiež schopné čítať a renderovať formát 2.01, ale nebude možné aplikovať nové vlastnosti.

SoundFont 2.04 (nikdy nebola verzia 2.02 alebo 2.03) bola predstavená v roku 2005 so zariadením Sound Blaster X-Fi. Formát 2.04 pridal podporu pre 24-bitové vzorky. Formát 2.04 je obojsmerne kompatibilný s formátom 2.01, takže syntezátory, ktoré sú schopné

reprodukovat iba formát 2.0 alebo 2.01, by automaticky vykreslili zvuky nástrojov pomocou 24-bitových vzoriek na 16-bitovú presnosť.

SoundFont je registrovaná ochranná známka spoločnosti E-mu Systems, Inc. a spoločnosť Digital Sound Factory získala exkluzívnu licenciu na preformátovanie a správu historického obsahu SoundFont [10].

Funkcionalita

MIDI súbory v sebe neobsahujú žiadne zvuky, obsahujú však inštrukcie na ich prehrávanie. Na prehrávanie súborov MIDI v syntezátoroch sa používajú MIDI syntetické vzorky na nahrávanie nástrojov alebo zvukov uložených v súbore alebo v čipe. Syntezátory kompatibilné so systémom SoundFont umožňujú záujemcom používať banky SoundFont s vlastnými vzorkami na prehrávanie hudby.

V súboroch SoundFontu sú samozrejme uložené zvukové vzorky. Každá zvuková vzorka má priradenú notu, ktorá by mala zodpovedať jeho výške. To neplatí pre zvuky bicích. Zvuk požadovanej výšky potom vznikne zrýchlením alebo spomalením vzorky nástroja. Nástroj môže obsahovať viac vzoriek, zvuk je potom kvalitnejší. Okrem zvukových vzoriek obsahuje aj napríklad obálku hlasitosti, hlasitostné vibrato, frekvenčné vibrato, je taktiež možné zvuk doladiť alebo pridať dozvuk alebo chorus [12].

Banky SoundFont môžu vyhovovať štandardným zvukovým setom, ako je napríklad General MIDI, taktiež sa môžu použiť iné definície zvukových súborov.

2.5 Cloudové úložisko iCloud

Aplikácia *The Piano Master* ukladá užívateľmi vytvorené lekcie do cloudového úložiska iCloud. V nasledujúcom texte je popísané cloudové úložisko iCloud ako aj framework CloudKit.

CloudKit

CloudKit je framework pre ukladanie a synchronizáciu dát medzi viacerými Apple ID účtami. CloudKit využíva prístup k iCloud serverom a umožňuje prácu na operačných systémoch iOS a OS X. CloudKit pracuje s kontajnermi (*CKContainer*). Každý kontajner má jednu public, jednu private databázu a shared databázu. Každá aplikácia má vlastný kontajner. Množstvo privátnych databáz sa zhoduje s počtom užívateľov aplikácie. Každý užívateľ má vlastnú privátnu databázu. Pre správu CloudKitu v aplikácii sa využíva CloudKit Dashboard. CloudKit Dashboard je nástroj pre správu dát, schém a rolí [18].

Na developerskej konferencii poriadanej spoločnosťou Apple v roku 2015 bol predstavený *REST2 API CloudKit JS*, ktorý umožňuje vytvoriť webovú službu, ktorá dokáže komunikovať s iCloud servermi a využíva k tomu CloudKit databázu [14].

iCloud

iCloud je cloudové úložisko od spoločnosti Apple. Umožňuje ukladať aplikačné dáta aj dokumenty a pristupovať k nim na všetkých Apple zariadeniach a na webe [3].

Služba funguje na princípe synchronizácie všetkých zariadení ktoré spadajú pod jeden Apple ID účet. Služba je poskytovaná zadarmo, ale je obmedzená množstvom pamäte. Za poplatok je možné navýšiť kapacitu pamäte. Nedá sa využiť pre synchronizáciu dát medzi viacerými užívateľmi, z dôvodu rozdielných Apple ID účtov. Základom každej iCloud

aplikácie je ubiquity kontajner. Dáta využívajúce iCloud sú najprv umiestnené do ubiquity kontajneru a následne sú tieto dáta poslané do iCloudu. V prípade zmien na iCloudu sú dáta poslané do ubiquity kontajneru. Smyth [18] popísal využitie služby iCloud pre nasledujúce typy ukladania:

- Key-value storage - vhodný pre ukladanie malých dát.
- iCloud Drive - alternatíva k službám typu Dropbox.
- iOS Backup - vhodný pre zálohu celého zariadenia.
- Core Data with iCloud - vhodný pre zálohu dát lokálneho úložiska Core Data.
- Document storage - vhodný pre ukladanie dokumentov a aplikačných dát.

Pri vývoji iOS aplikácií sa pre využitie iCloudu používa framework CloudKit. CloudKit teda zaistuje rozhranie pre komunikáciu danej aplikácie a iCloudu [2].

2.6 Core Data

Moja aplikácia umožňuje vytvárať zoznamy obľúbených lekcí a piesní, táto funkcionality je zabezpečená pomocou knižnice Core Data. V tejto kapitole je bližšie popísaná knižnica Core Data.

Core Data

Význam Core Data spočíva v uložení dát na disku daného zariadenia. V Apple dokumentácii [6] je o Core Data napísané nasledovné. Core Data je prirodzenou knižnicou platformy iOS a MacOS pre riešenie modelovej vrstvy aplikácie. Apple uvádza, že využitie tejto technológie programátorom ušetrí 50-70% kódu. Core Data má množstvo funkcií, ktoré vývojárom uľahčujú prácu a zamedzujú zbytočnej duplikácii kódu. Nižšie sú uvedené niektoré z nich pre prehľad.

- Údržba propagácie zmien v hierarchii dát a sledovanie konzistencie vzťahov medzi dátami.
- Zabudované sledovanie zmien v uložených dátach a management zmien s históriou krokovania späť a dopredu.
- Nahrávanie dát v okamihu potreby a vytvorenie kópie a zápis dát až v prípade potreby.
- Automatická podpora pre ukladanie objektov v externých dátových úložiskách.
- Možnosť prepojenia dátovej vrstvy cez controller až do vrstvy užívateľského rozhrania a propagácia zmien v dátach.
- Zhlukovanie, filtrovanie a organizácia dát v pamäti a v užívateľskom rozhraní.

Core Data môže serializovať dáta do formátu XML, do binárneho súboru alebo do SQLite. Core Data potom komunikuje priamo s SQLite, programátor teda nemusí využívať priamo jazyk SQL. Výhodou tohto frameworku je, že je zdarma a má podporu priamo v Xcode.

Core Data Stack

Core Data Stack[4] - prostredník medzi objektami v aplikácii a externým dátovým úložiskom.

- Managed Object Model - schéma popisujúca entity, ich atribúty a vzťahy
- Persistent store coordinator - je zodpovedný za získavanie dát do Managed Object Context a následne navrátenie do persistentného úložiska (NSPersistentStore)
- Managed Object Context - kópia získaných objektov z persistentného úložiska, s ktorými sa dá manipulovať.

Kapitola 3

Návrh výslednej aplikácie

3.1 Prieskum cieľovej skupiny

Pri návrhu výslednej aplikácie je dobré mať na pamäti cieľovú skupinu užívateľov. Aplikácia *The Piano Master* očakáva užívateľov, ktorí sa chcú učiť hrať na klávesové nástroje. Prípadne očakáva užívateľov, ktorí chcú vytvárať lekcie hry na klávesové nástroje s možnosťou zdieľať tieto lekcie s komunitou a to všetko prostredníctvom inteligentného mobilného telefónu. Výhodou teda je, že inteligentný telefón vlastní väčšina populácie a užívateľské rozhranie by malo spĺňať čitateľný a jednoduchý design. K tomuto bodu smeruje aj fakt, že výučba hry na klávesové nástroje nie je vekovo obmedzená a skúšať učiť hru na klavír sa môže ako aj 8-ročný žiak základnej školy, tak aj 70-ročná staršia pani na dôchodku.

3.2 Vytýčenie cieľov

Pre svoju aplikáciu *The Piano Master* som si vytýčil nasledujúce ciele:

- Aplikácia by mala mať taký design, aby užívatelia zcela jasne vedeli, kde sa v aplikácii nachádzajú a ako sa dostanú späť.
- Aplikácia by mala slúžiť na výučbu hry na klávesové nástroje formou základných lekcí, a následne formou zložitejších piesní.
- Užívateľ by mal z aplikácie *The Piano Master* pochopiť akú náročnú lekciiu si má vybrať podľa vlastných zručností a skúseností s hrou na klavír.
- Aplikácia by mala umožňovať meniť tempo jednotlivých lekcí.
- Aplikácia by mala umožniť filtrovať hranie pravej alebo ľavej ruky, ako aj hranie oboch rúk naraz.
- Mala by umožniť užívateľovi cvičiť ťažšie pasáže z lekcí a piesní formou opakovania danej pasáže.
- Aplikácia by mala umožniť užívateľskú tvorbu lekcí.
- Aplikácia by mala umožniť sprístupniť vytvorené lekcie všetkým užívateľom.
- Taktiež by malo byť dostupné zdieľať prostredníctvom aplikácie jednotlivé lekcie na sociálnych sieťach.

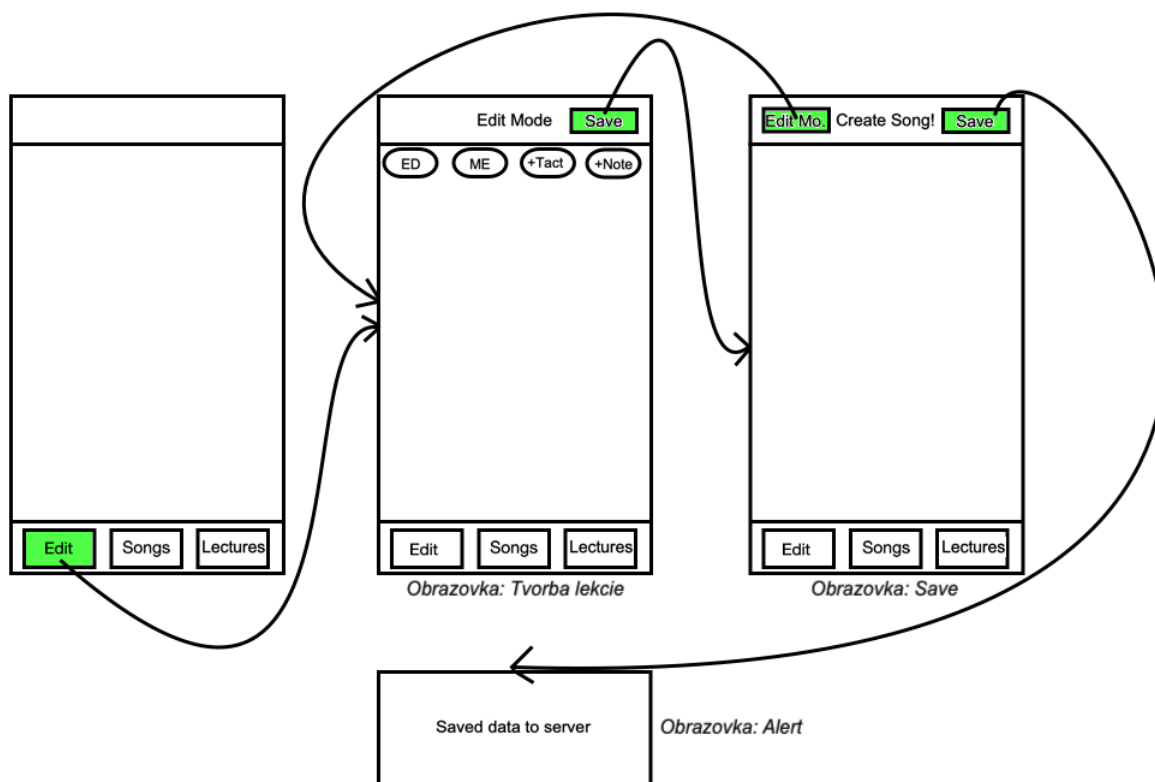
3.3 Návrh užívateľského rozhrania

Všetky obrazovky aplikácie a prechody medzi nimi

Aplikácia obsahuje 3 základné časti, ktoré sú dostupné z takmer každej časti aplikácie, a to formou menu v dolnej časti. V tejto časti sú popísané hlavné obrazovky aplikácie a prechody medzi nimi. Bližší popis návrhu niektorých významných obrazoviek bude vysvetlený v ďalšej časti tejto kapitoly.

Prvá položka v hlavnom menu - Edit

Prvá položka v menu aplikácie je ViewController, ktorého úloha spočíva vo vytváraní lekcí užívateľom, čiže editačný mód aplikácie. Po vytvorení lekcie má užívateľ možnosť popísať a kategorizovať túto lekcii kliknutím na tlačidlo „Save“ v pravom hornom rohu. Po kliknutí na toto tlačidlo sa užívateľ dostane na ďalšiu obrazovku, ktorej úlohou je špecifikovať vlastnosti vytvorenej lekcie/piesne. Pokiaľ sa chce užívateľ vrátiť k editácii lekcie, má možnosť tak urobiť kliknutím na ľavé horné tlačidlo „Edit Mode“. Po vyplnení špecifikácie lekcie potvrdí užívateľ tieto informácie kliknutím na tlačidlo „Save“ v pravom hornom rohu. Ak sú údaje vyplnené správne, tak sa po splnení tejto akcie lekcia uloží na server, a tým sa sprístupní ostatným užívateľom aplikácie. Táto časť aplikácie je zobrazená na Obrázku 3.1.



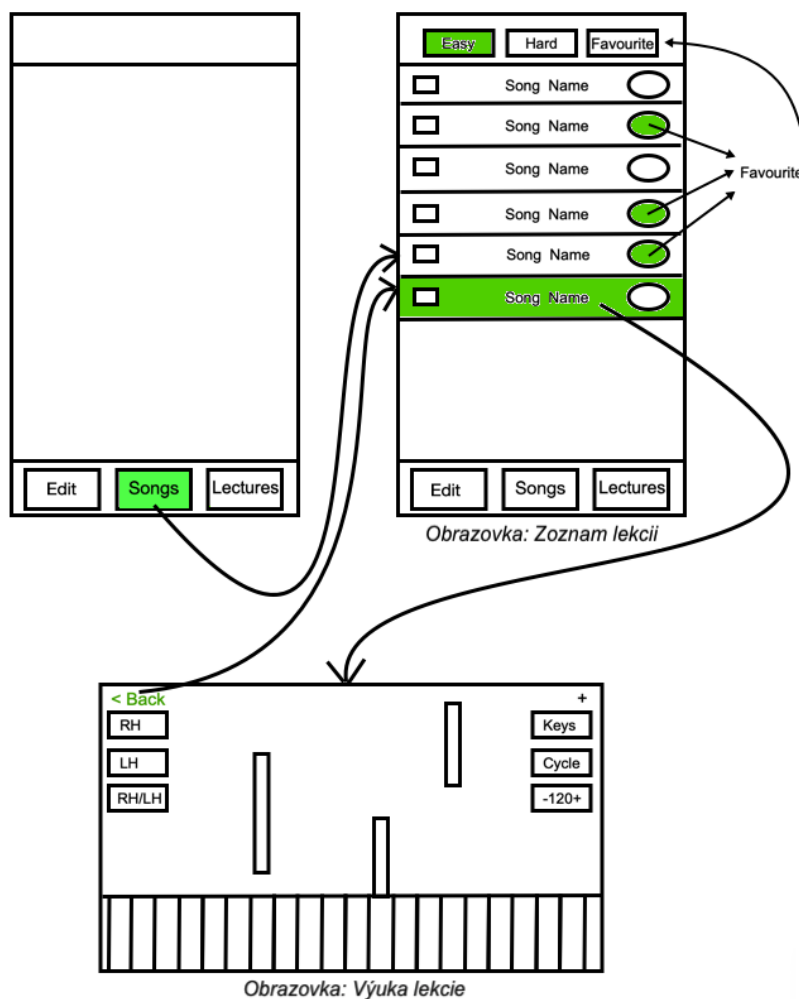
Obrázek 3.1: Prvá časť aplikácie - Edit.

Druhá položka v hlavnom menu - Songs

V druhej časti menu v aplikácii sa nachádza výukový mód aplikácie. Obrazovka obsahuje tri segmenty, ktorých štruktúra je rovnaká avšak obsah je rozdielny.

- Prvý segment tvoria základné lecie výuky hry ako napríklad stupnice.
- Druhý segment tvoria zložitejšie lecie výuky - rôzne známe, ale aj menej známe piesne.
- Tretí segment tvorí zoznam obľúbených lekcí a piesní. Tento zoznam si vytvára každý užívateľ samostatne.

Každý zo segmentov je tvorený zoznamom položiek (lekcií). Po kliknutí na položku sa užívateľ dostane na obrazovku, ktorej úlohou je prezentovať konkrétnu lekciiu. Hlavnou časťou tohto ViewControlleru je klaviatúra, na ktorej sa premietajú jednotlivé noty piesne. Po kliknutí na tlačidlo „+“ v pravom hornom rohu obrazovky, sa zobrazia jednotlivé možnosti spravovania danej lekciiu. Tlačidlo „<- Back“ slúži k navráteniu sa k zoznamu položiek. Táto časť aplikácie je zobrazená na Obrázku 3.2.



Obrázek 3.2: Druhá časť aplikácie - Songs.

Tretia položka v hlavnom menu - Lectures

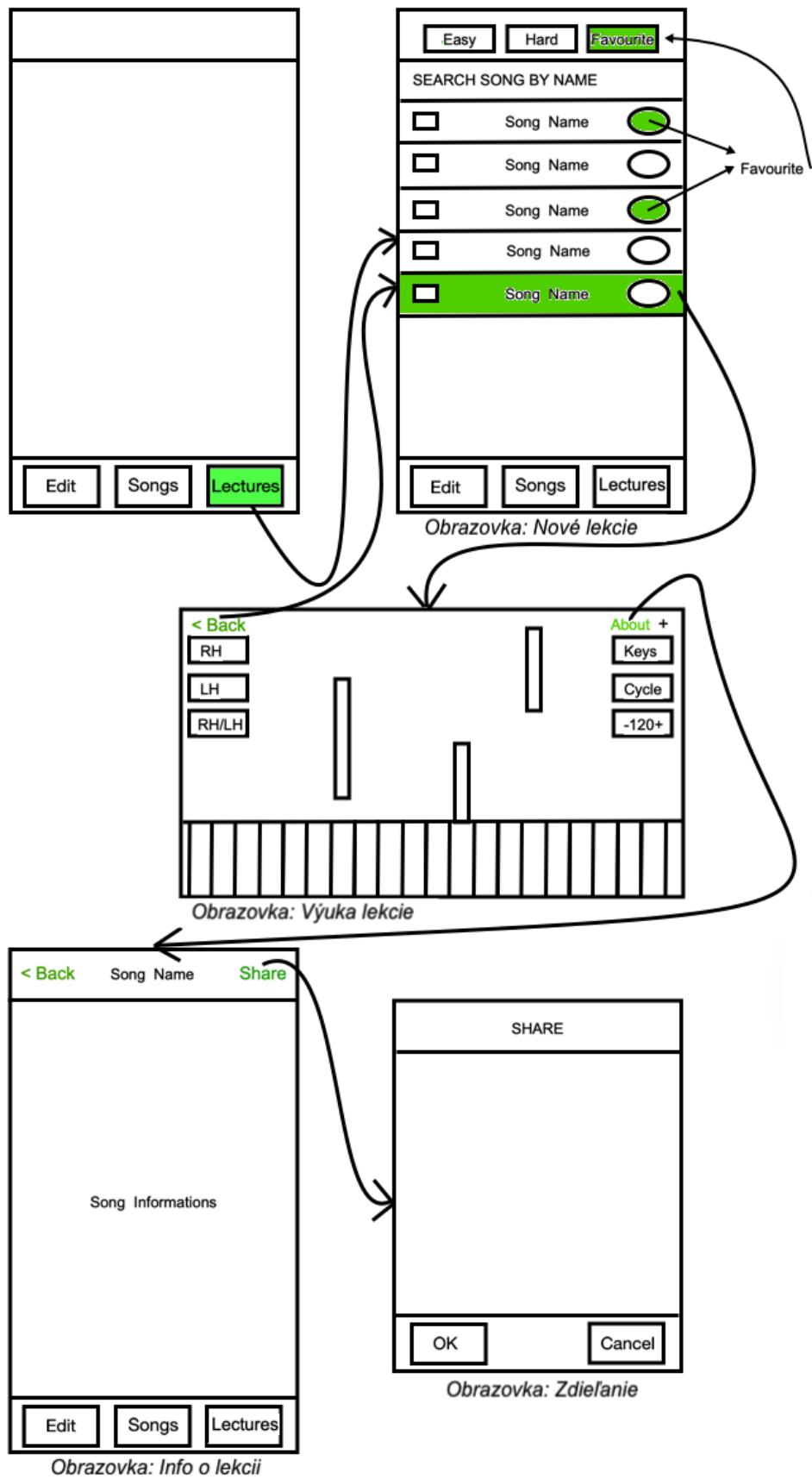
V tretej časti menu v aplikácii sa nachádzajú všetky lekcie vytvorené všetkými užívateľmi aplikácie. Obrazovka taktiež obsahuje tri segmenty.

- Prvý segment tvoria jednoduché lekcie.
- Druhý segment tvoria zložitejšie lekcie.

Tieto dva segmenty obsahujú zoznamy lekcí, ktoré sa pri pripojení k internetu automaticky v reálnom čase obnovujú všetkým užívateľom aplikácie.

- Tretí segment tvorí zoznam užívateľom uložených lekcí.

Užívateľ má možnosť vyhľadávať spomedzi všetkých lekcí pomocou vyhľadávacieho poľa v hornej časti obrazovky. Po kliknutí na lekcii sa zobrazí prezentačný mód konkrétnej lekcie, ako tomu bolo aj v druhej časti menu vo výukovom móde. Po kliknutí na tlačidlo „+“ v pravom hornom rohu obrazovky sa zobrazia jednotlivé možnosti spravovania danej zdieľanej lekcie. Tlačidlo „< Back“ slúži k navráteniu sa k zoznamu položiek. V pravom hornom rohu sa nachádza tlačidlo „About“. Po kliknutí na toto tlačidlo sa zobrazí ďalšia obrazovka, na ktorej sa nachádzajú špecifikácie tej danej lekcie. Tlačidlo „< Back“ spôsobí navrátenie sa k prezentácii lekcie. Tlačidlo „Share“ v pravom hornom rohu umožňuje zdieľať danú lekcii na sociálnych sieťach. Stlačenie tohto tlačidla spôsobí vygenerovanie obrázku s kategóriou a názvom lekcie. Následne môže užívateľ pridať komentár k tomuto príspevku, ktorý sa po potvrdení zdieľa na sociálnej sieti. Táto časť aplikácie je zobrazená na Obrázku 3.3.



Obrázek 3.3: Tretia časť aplikácie - Lectures.
20

Významné čiastkové obrazovky

V tejto sekcii si popíšeme návrh niektorých významných obrazoviek.

Obrazovka: Tvorba lekcie

V nasledujúcom texte si ukážeme detail na obrazovku „Tvorba lekcie“ prezentovanú na Obrázku 3.1 ako jednu z čiastkových obrazoviek v časti menu s názvom „Edit“.

Návrh systému obrazovky „Tvorba lekcie“ spĺňa tieto základné požiadavky:

- Systém je ľahký na pochopenie.
- Systém je schopný kvôli prehľadnosti zobrazíť celú klaviatúru s väčším množstvom nôt.
- Systém je schopný adaptovať sa na väčšie množstvo taktov, systém je schopný pridať určitý počet taktov.
- Systém umožňuje pridávanie nôt.
- Systém je schopný upraviť a presúvať väčšie množstvo nôt naraz.

Z vyššie uvedených bodov bolo usúdené, že „pracovná plocha“ - klaviatúra s notami - musí byť kvôli prehľadnosti čo najväčšia, a teda tvorí značnú časť obrazovky. Ovládacie prvky (pridávanie nôt, pridávanie taktov, editácia taktov, multi-editácia taktov) tvoria hornú časť obrazovky.

Kvôli prehľadnosti sú na pracovnej ploche vizuálne odlíšené hranice taktov a ich menších častí a kláves na klaviatúre. Vytvorená nota je taktiež vizuálne odlíšená. Z tohoto dôvodu je každá nota tvorená jedným View. Nota je schopná na podnet užívateľa zväčšovať svoju dĺžku a taktiež je užívateľovi umožnené jednoduchým „chytením“ noty presúvať túto notu po pracovnej ploche.

Užívateľ má možnosť presúvať sa jednoduchým spôsobom po celej pracovnej ploche, čo je docielené využitím ScrollView a zoomu. Tým pádom je možné zoomovať a pohybovať sa v horizontálnom aj vertikálnom smere po pracovnej ploche.

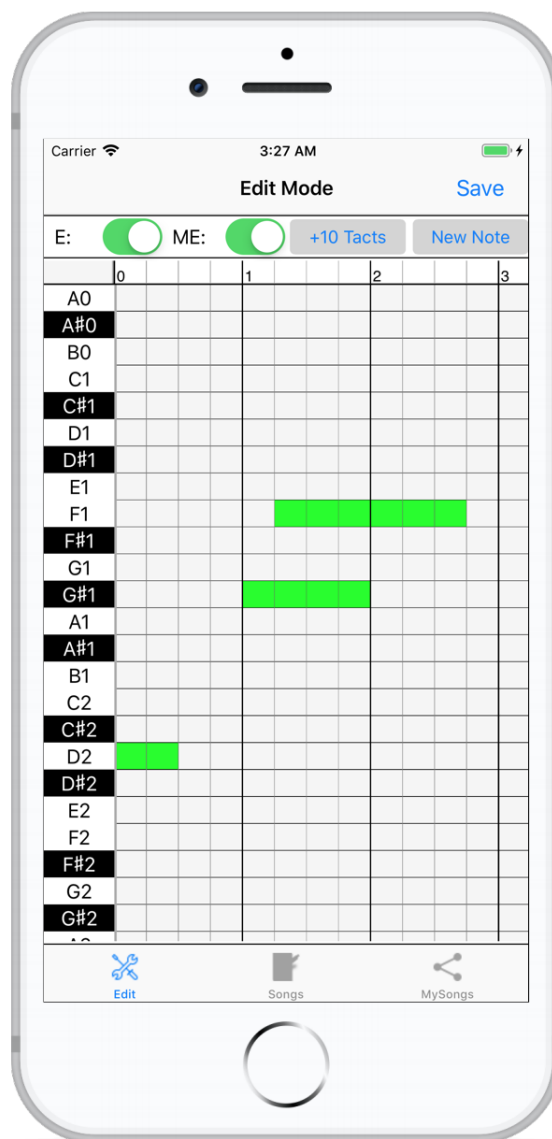
Výslednú podobu tejto obrazovky popisuje Obrázok 3.4.

Obrazovka: Výuka lekcie

V nasledujúcom texte si ukážeme detail na obrazovku „Výuka lekcie“ prezentovanú na Obrázku 3.2 ako jednu z čiastkových obrazoviek v časti menu s názvom „Songs“.

Návrh systému obrazovky „Výuka lekcie“:

- Systém je schopný vizuálne reprezentovať lekciiu.
- Systém umožňuje extrahovať z MIDI súboru dôležité informácie, ktoré sa zobrazujú na obrazovke.



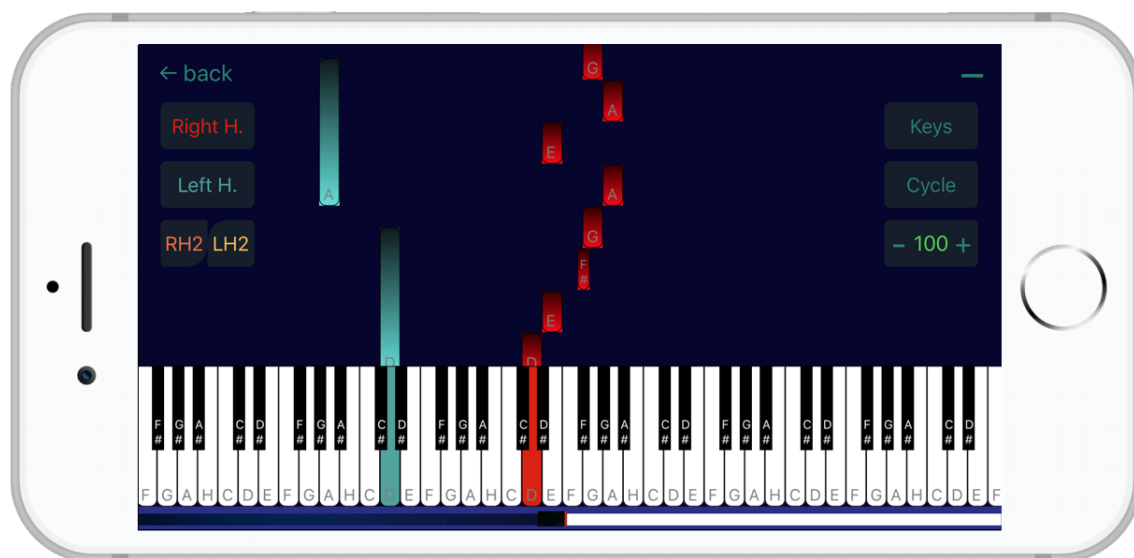
Obrázek 3.4: Výsledná podoba obrazovky: Tvorba lekcie.

- Systém vyobrazuje časovú os lekcie s možnosťou nastavenia, kde presne sa chce užívateľ v lekcii nachádzať.
- Systém umožňuje užívateľovi filtrovať, ktorú ruku chce cvičiť. A to aj pri hre 2 hráčov naraz.
- Systém taktiež umožňuje hru oboch rúk. Ale aj hru 2 hráčov naraz so 4 rukami.
- Systém je schopný umožniť zobraziť na klaviatúre notové označenie jednotlivých kláves. Označenie hodnoty nôt je možné zobraziť aj na grafických zobrazeniach premietajúcich sa nôt.
- Systém spĺňa požiadavky na vytváranie cyklov pri zložitejších pasážach lekcie, čo umožňuje užívateľovi nacvičiť ťažší úsek opakovaním tohto úseku viackrát po sebe.

- Systém je schopný umožniť užívateľovi nastaviť tempo lekcie podľa vlastných predstáv.

Nakoľko táto obrazovka je kľúčová vo výukovom móde aplikácie a je zjavné, že pri cvičení lekcí, ale aj učení sa nových piesní strávi užívateľ najviac času na tejto obrazovke, je zobrazené na obrazovke iba to najdôležitejšie ako je klaviatúra s premietajúcimi sa notami a časovou osou lekcie prípadne piesne. Na túto obrazovku je použitá východzia možnosť Landscape módu aplikácie. Pokiaľ je však potrebné nastaviť jednotlivé aspekty lekcie podľa aktuálnych potrieb užívateľa, je dostupná možnosť zobrazenia akéhosi „lokálneho“ menu obrazovky. Na tento účel je sprístupnené ešte jedno tlačidlo na zobrazenie práve týchto nastavovacích tlačidiel. Aby bolo možné cvičiť aj počas toho ako je zobrazené lokálne menu výukovej obrazovky, je potrebné tieto nastavovacie tlačidlá umiestniť na pravú a ľavú stranu obrazovky v Landscape móde aplikácie. Toto rozloženie jednotlivých prvkov umožňuje súčasne komfortne cvičiť, ale aj nastaviť vlastnosti lekcie.

Výslednú podobu tejto obrazovky popisuje Obrázok 3.5.



Obrázok 3.5: Výsledná podoba obrazovky: Výuka lekcie.

3.4 Návrh komunikácie mobilnej aplikácie

Vývojár mobilnej aplikácie, v ktorej je potrebné uchovávať nejaké dáta, prípadne je potrebné tieto dáta zdieľať, má niekoľko možností ako s dátami pracovať. Najjednoduchší spôsob je ukladať tieto dáta do lokálnej pamäti. Ďalšou možnosťou je využitie nejakej databázy, s ktorou bude mobilná aplikácia komunikovať. Je tu tiež možnosť vytvoriť si serverovú aplikáciu, ktorá bude tieto služby obstarávať a aplikácii bude ponúkať prístup napríklad cez REST API. Na trhu existuje už dlhšiu dobu platforma Google Firebase, ktorá ponúka množstvo nástrojov, ktoré pomáhajú vývojárom mobilných aplikácií riešiť problémy tohto druhu.

Návrh zdieľania vytvorených lekcí pomocou služby iCloud

Pri návrhu komunikácie aplikácie *The Piano Master* je potrebné brať do úvahy fakt, že aplikácia pracuje v reálnom čase a teda, že užívateľmi vytvorené lekcie musia byť dostupné všetkým užívateľom hneď po vytvorení danej lekcie. Z tohto dôvodu je vhodné pre aplikáciu *The Piano Master* použiť službu iCloud - cloudové úložisko od spoločnosti Apple. Toto riešenie umožňuje ukladať aplikačné dáta a dokumenty a pristupovať k nim na všetkých Apple zariadeniach, ktoré majú nainštalovanú aplikáciu *The Piano Master*.

V aplikácii *The Piano Master* je využitá trieda typu *CKRecord*. Je to základná trieda, ktorá sa využíva pre spracovanie dát v CloudKite. Trieda *CKRecord* reprezentuje záznam v dátovom modeli. V aplikácii môže byť využité ľubovoľné množstvo záznamov, pričom každý záznam zodpovedá inému typu informácie, ktoré je potrebné uchovať. V rámci jedného záznamu je možné definovať jedno alebo viac polí, z ktorých každé pole má názov a hodnotu údajov. Záznamy môžu obsahovať jednoduché typy údajov, ako sú textové reťazce a čísla, alebo zložitejšie typy. Zoznam typov údajov, ktoré je možné využiť je nasledovný:

- *NSString* - Slúži na ukladanie relatívne malého množstva textu. I keď samotné reťazce môžu mať akúkoľvek dĺžku, na uloženie väčšieho množstva textu je vhodné použiť *CKAsset*.
- *NSNumber* - Slúži na uloženie všetkých číselných informácií, vrátane celých čísel a čísel s desatinnou čiarkou.
- *NSData* - Slúži na ukladanie bytových informácií.
- *NSDate* - Slúži na ukladanie dátumových a časových informácií v prístupnej forme.
- *NSArray* - Slúži na ukladanie polí ostatných objektov.
- *CLLocation* - Slúži na ukladanie informácií o geografickej polohe.
- *CKAsset* - Slúži na ukladanie diskových dát, ako sú napríklad obrázky.
- *CKRecord.Reference* - Vytvára prepojenie medzi záznamami. Obsahuje ID cieľového záznamu.

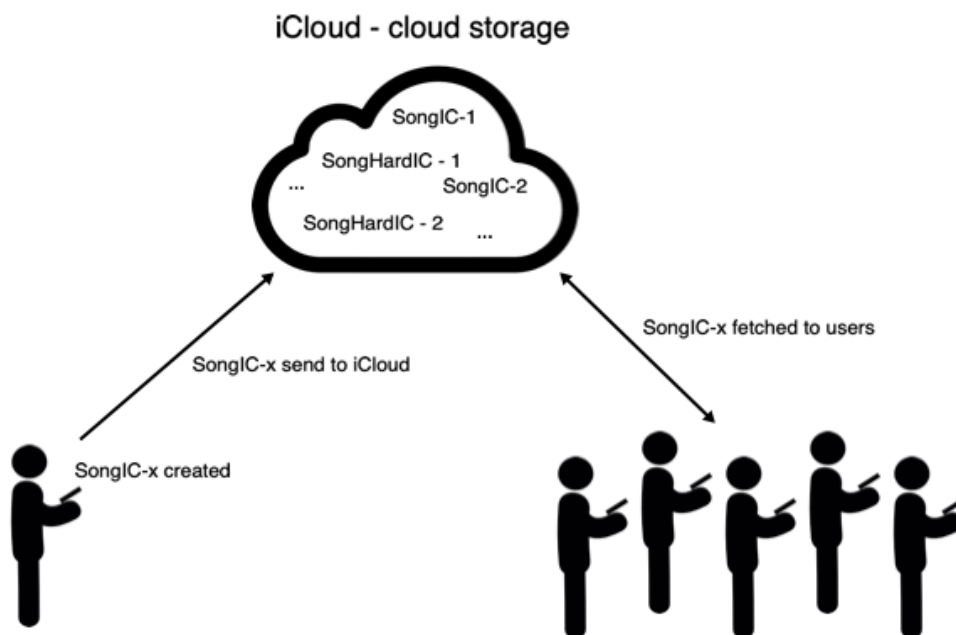
V aplikácii *The Piano Master* sa využíva trieda s názvom *SongIC*. Trieda *SongIC*, ktorá reprezentuje záznam obsahuje informácie o jednej vytvorenej lekcií. Atribúty triedy *SongIC* sú vo formáte: názov, dátový typ a popis atribútu.

- **authorNameIC (NSString)** - meno autora lekcie
- **aboutEPSongIC (NSString)** - popis lekcie
- **categoryIC (NSString)** - kategória, do ktorej spadá lekcia
- **idIC (NSNumber)** - identifikačné číslo lekcie
- **songMidiArrayIC (NSString)** - MIDI informácia lekcie
- **imageIC (NSString)** - názov obrázku lekcie
- **nameIC (NSString)** - názov lekcie

- `numOfPlayingIC` (`NSNumber`) - počet prehratí danej lekcie
- `songNameIC` (`NSString`) - názov piesne lekcie

Aby bolo možné v aplikácii vyhľadávať lekcie pomocou názvu lekcie v oboch kategóriách (ľahšie lekcie, ťažšie lekcie), sú využité dva záznamy s rovnakými atribútmi popísanými vyššie. Líšia sa však v kategórii, do ktorej spadajú. Keď chce užívateľ vyhľadávať v ľahších lekciách, môže tak urobiť, pričom mu aplikácia poskytne výsledky vyhľadávania iba z ľahších lekcií. Keď chce vyhľadávať naopak v ťažších lekciách, tak mu aplikácia poskytne výsledky z ťažších lekcií. Táto funkcionality je dosiahnutá dvomi druhmi záznamov. Prvý z nich má názov *SongIC* a druhý *SongHardIC*. Trieda *SongHardIC* obsahuje rovnaké atribúty ako trieda *SongIC*.

Keď teda užívateľ vytvorí lekciu, tak mu aplikácia poskytne možnosť vyplniť dodatočne informácie o lekci, ako napríklad meno autora, popis lekcie atď. Následne po potvrdení všetkých údajov a vybratí kategórie lekcie sa vytvorí objekt, ako inštancia triedy *SongIC* alebo *SongHardIC* podľa toho, do ktorej kategórie lekcia spadá. Tento objekt sa následne uloží ako záznam do verejnej databázy na iCloud. Všetci ostatní užívatelia majú teda v reálnom čase prístup k tejto novo vytvorenej lekci. Proces zdieľania lekcií je zobrazený na Obrázku 3.6. Užívatelia môžu filtrovať a vyhľadávať lekcie pomocou názvu priamo v aplikácii *The Piano Master*.



Obrázek 3.6: Zdieľanie lekcií.

Návrh ukladania obľúbených lekcií pomocou Core Data

Ďalšou vlastnosťou aplikácie *The Piano Master* je vytváranie vlastných zoznamov obľúbených piesní, ako aj ukladanie potrebných zdieľaných lekcií do vlastného zoznamu. Nakoľko je potrebné, aby vytvorený uložený zoznam bol dostupný iba na konkrétnom zariadení, tak vhodnou voľbou na uloženie týchto zoznamov je technológia Core Data, ktorej výhodou

je, že je zdarma a má podporu priamo v Xcode. Ide teda o uloženie dát na disku daného zariadenia.

V Aplikácii *The Piano Master* sa využíva entita s názvom *EPFavouriteSongItemCD*. Entita *EPFavouriteSongItemCD* slúži k uloženiu obľúbených lekcií do persistentného úložiska konkrétneho zariadenia na ktorom pracuje užívateľ. Táto entita obsahuje nasledujúce atribúty (vo formáte názov, dátový typ a popis atribútu):

- **authorNameCD (String)** - meno autora lekcie
- **aboutEPSongCD (String)** - popis lekcie
- **categoryCD (String)** - kategória do ktorej spadá lekcia
- **idCD (Int16)** - identifikačné číslo lekcie
- **songMidiArrayCD (String)** - MIDI informácia lekcie
- **imageCD (String)** - názov obrázku lekcie
- **nameCD (String)** - názov lekcie
- **numOfPlayingCD (Int16)** - počet prehratí danej lekcie
- **songNameCD (String)** - názov piesne lekcie

Užívateľ má možnosť vytvoriť a upravovať si zoznam obľúbených lekcií, ktoré sú následne dostupné aj počas režimu offline, keď užívateľ nemá prístup k internetovému pripojeniu.

Podobne je vyriešené aj ukladanie stupníc a zložitejších piesní do persistentného úložiska zariadenia. Na uloženie týchto dát sa využíva entita s názvom *FavouriteSongItem*. Entita *FavouriteSongItem* obsahuje nasledujúce atribúty (vo formáte názov, dátový typ a popis atribútu):

- **favouriteSongItemId (Int32)** - identifikačné číslo piesne
- **favouriteSongItemImage (String)** - názov obrázku piesne
- **favouriteSongItemLbl (String)** - názov piesne
- **favouriteSongItemSwitchStatus (Boolean)** - identifikácia uloženia v zozname obľúbených piesni
- **favouriteSongItemSongName (String)** - názov MIDI súboru piesne

Ako je vidieť, aplikácia *The Piano Master* si neukladá do persistentného úložiska MIDI súbory jednotlivých stupníc a piesni, pretože tieto súbory sú v zariadení už uložené a teda nám stačí uložiť si iba názov MIDI súboru. Pri požiadavke interpretácie MIDI súboru poznáme názov súboru a teda vieme tento súbor dohľadať v zariadení.

Aplikácia *The Piano Master* teda umožňuje užívateľovi vytvoriť a upravovať si aj zoznam obľúbených stupníc a piesni.

3.5 Návrh logiky aplikácie, ako aplikácia preberá systém metodiky výuky

Dôležitým cieľom aplikácie *The Piano Master* je správne prebratie techniky výuky hry na klávesové nástroje. V aplikácii je umožnené prehrávanie piesni s vizuálnou aj zvukovou reprezentáciou nôt. Ako už bolo písané v podkapitole vyššie „Výuka hry na klávesové nástroje“, je dôležité si najprv vypočítať skladbu. Z MIDI súboru sa extrahujú všetky potrebné informácie o notách v danej skladbe. Následne je pomocou na internete voľne dostupného¹ *piano.sf2* súboru vygenerovaný piano zvuk jednotlivých nôt piesne, ktoré korešpondujú s notami popísanými v MIDI súbore. Pomocou týchto MIDI informácií je taktiež vygenerované vizuálne zobrazenie jednotlivých nôt ktoré sa počas prehrávania piesne premietajú na model klaviatúry.

Ďalšou dôležitou časťou správnej formy výuky hry na klávesové nástroje je hranie každej ruky zvlášť, ako bolo vysvetlené vyššie v tejto práci. V aplikácii *The Piano Master* je teda možné vyberať si v piesni ktorú ruku chceme cvičiť. Táto možnosť je docieľená tak, že z MIDI súboru sú v aplikácii rozlíšené 2 kanály (pri hre jedného človeka), prípadne 4 kanály (pri hre 2 ľudí), v ktorých každý kanál reprezentuje jednu ruku. Keď chce teda užívateľ cvičiť napríklad iba pravú ruku, začnú sa premietiť informácie len z jedného kanálu, ktorý reprezentuje pravú ruku. Taktiež zvuk piesne bude tvorený len časťou, ktorú hrá pravá ruka.

Ďalej je dôležité umožniť meniť tempo piesne pri výuke. Užívateľ je schopný zmeniť si v aplikácii *The Piano Master* tempo jednotlivých piesní. Táto funkcionálna je docieľená tak, že sa mení rýchlosť premietania nôt na klaviatúru vo výukovom móde aplikácie *The Piano Master*. Súčasne sa mení rýchlosť zvukovej reprezentácie piesne tak, aby odpovedala rýchlosti premietania nôt na klaviatúru. V aplikácii je možné spomaliť piesne, ale aj zrýchliť piesne, čo pomáha užívateľovi k lepšiemu precvičeniu jednotlivých lekcii.

Dôležitým aspektom v kapitole „Výuka hry na klávesové nástroje“ je taktiež umožniť žiakovi cvičiť opakovane niektoré ťažšie pasáže piesne. V aplikácii *The Piano Master* je táto možnosť vyriešená pomocou cyklenia jednotlivých pasáží, ktoré si určí užívateľ aplikácie. Užívateľ si určí začiatok cyklu kliknutím na tlačidlo na to určené. Keď chce užívateľ označiť koniec cyklu, stlačí tlačidlo na to určené. Aplikácia začne opakovane prehrávať časť piesne, ktorú užívateľ označil. Aplikácia *The Piano Master* taktiež umožňuje meniť rýchlosť tohto cyklu a umožňuje aj filtráciu rúk počas prehrávania cyklu.

3.6 Návrh komunitného zdieľania lekcii

Jedným z cieľov mobilnej aplikácie *The Piano Master* bola aj možnosť komunitného zdieľania jednotlivých lekcii na sociálnych sieťach.

Na tento účel sa v aplikácii *The Piano Master* používa trieda *UIActivityViewController*, ktorá dedí vlastnosti z triedy *UIViewController*. Trieda *UIActivityViewController* poskytuje štandardné služby, ktoré sú dostupné priamo z aplikácie. Medzi tieto štandardné služby patrí napríklad:

- Kopírovanie obsahu z jednej aplikácie do inej aplikácie.
- Odosielanie obsahu z aplikácie na sociálne médiá.

¹<https://www.flstudiomusic.com/2010/02/25-piano-soundfonts.html>

- Odosielanie obsahu aplikácie prostredníctvom e-mailu.
- Odosielanie obsahu aplikácie prostredníctvom SMS.
- Zdieľanie príslušného obsahu aplikácie v zdieľaných albumoch.
- Taktiež môže byť definované vlastné špecifické chovanie *UIActivityViewController*.

V aplikácii *The Piano Master* sa využíva odoslanie obsahu z aplikácie na sociálne médium *Facebook*. Užívateľ musí byť prihlásený na *Facebooku* na zariadení, na ktorom je aplikácia *The Piano Master* spustená. Pri zdieľaní konkrétnej lekcie sa vytvorí obrázok s názvom lekcie a s kategóriou, do ktorej lekcia spadá (pre ľahšie vyhľadanie v aplikácii). Ďalej užívateľ môže pridať svoj vlastný popis, názor na lekciu, ktorú chce zdieľať s komunitou. Užívateľia následne môžu na sociálnej sieti *Facebook* komentovať a lajkovať zdieľaný príspevok. Nižšie pod týmto textom je zobrazená implementácia funkcie *share()*, ktorá slúži na zdieľanie obsahu pomocou *UIActivityViewController*.

```
// Sharing with UIActivityViewController
func share() {

    let memedImage = self.textToImage(
        drawText: "Category: \(ePsong.category)",
        atPoint: CGPoint(x: 10, y: 70),
        drawText2: "Search: \(ePsong.songName)",
        atPoint2: CGPoint(x: 10, y: 468),
        inImage: UIImage(named: "piano1-smaller")!)

    let activityController = UIActivityViewController(
        activityItems: [memedImage],
        applicationActivities: nil)

    activityController.completionWithItemsHandler = {
        (activityType: UIActivity.ActivityType?,
         completed: Bool,
         returnedItems: [Any]?,
         error: Error?) in

        if !completed {
            return
        }
        return
    }

    self.present(activityController, animated: true, completion: nil)
}
```


Kapitola 4

Implementácia mobilnej aplikácie pre podporu výuky hry na klávesové nástroje

V tejto časti je popísaná samotná implementácia so zameraním na podstatné prvky celej aplikácie ako sú implementácia užívateľského rozhrania, implementácia generovania zvuku, implementácia importu a exportu skladieb.

4.1 Implementácia užívateľského rozhrania

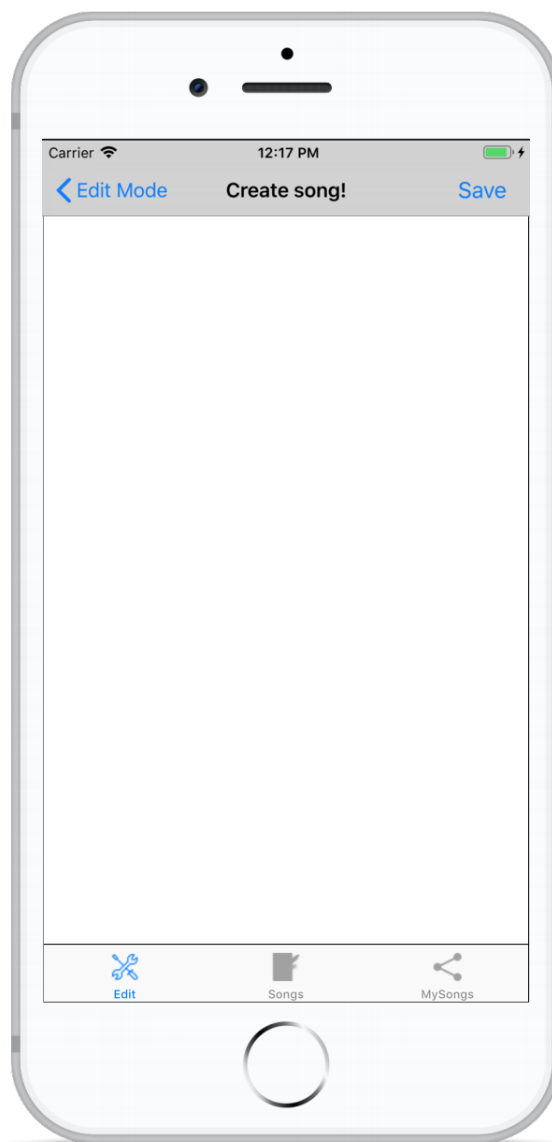
Aplikácie bola implementovaná podľa užívateľského rozhrania, ktorého návrh bol popísaný v kapitole 4.2.

Užívateľské rozhranie je vytvorené vo vývojovom prostredí *Xcode*, ktoré ponúka implementáciu pomocou *Storyboardov*. Všetky obrazovky aplikácie, okrem jednej (Výuka lekcie) využívajú spoločné *Tab Bar menu*. *Tab Bar menu* môžeme vidieť vyobrazené na Obrázku 4.1. *Tab Bar menu* je spodné prepínacie menu, ktoré slúži na rýchle prepínanie medzi hlavnými obrazovkami aplikácie. Na Obrázku 4.1 môžeme taktiež vidieť navigačné tlačidlá. O tieto tlačidlá sa stará *Navigation Controller*.

Zobrazenie dát v `UITableViewController`

Niektoré dáta v aplikácii je potrebné nasádzať do tabuľky. V Interface Builderi na tento účel využijeme *TableView*. K *TableView* musí byť vytvorený prislúchajúci controller. Na to nám poslúži *UITableViewController*.

UITableViewController obsahuje niekoľko metód, ktoré sú potrebné na správne zobrazenie obsahu v tabuľke *TableView*. Jednou z metód je *numberOfSections*. Táto metóda vracia počet sekcií v tabuľke *TableView*. V aplikácii *The Piano Master* je vo všetkých tabuľkách *TableView* dostačujúca 1 sekcia. Ďalšou povinnou metódou je *numberOfRowsInTableView*. Metóda *numberOfRowsInTableView* vracia počet riadkov tabuľky *TableView*. Napríklad v aplikácii *The Piano Master* v zozname piesní značí každý riadok tabuľky *TableView* jednu pieseň. Ďalej je pomocou tabuľky *TableView* v aplikácii *The Piano Master* riešený aj popis lekcí. Každá lekcia obsahuje nejaké informácie, ako napríklad názov lekcie, meno autora, kategóriu atď. Všetky tieto informácie sú uložené do tabuľky *TableView*, kde každý riadok tabuľky tvorí jednu informáciu o lekcii.



Obrázek 4.1: Tab Bar menu + Navigation Bar.

4.2 Využitie MIDI súborov a import a export skladieb

Vo výukovom móde aplikácie má užívateľ možnosť vybrať si z ľahších lekcí, ako sú napríklad stupnice, alebo z ťažších lekcí, ktoré sú tvorené piesňami. Nosnou časťou týchto lekcí je MIDI súbor, v ktorom sú uložené informácie o lekciami. V aplikácii *The Piano Master* sa využíva formát MIDI práve z toho dôvodu, že to je veľmi často využívaný a rozšírený formát v hudobnej sfére, a môže byť výhodou aj do budúcnosti v prípade rozšírenia aplikácie o nové funkcionality, prípadne komunikáciu s inými zariadeniami, aplikáciami, ale aj hudobnými nástrojmi. Prípadné budúce prenášanie MIDI súboru do iného zariadenia alebo aplikácie podporujúce MIDI súbory, sa tak tieto lekcie a piesne môžu ľahko stať kompatibilnými v širokej sfére. MIDI súbory výukovej časti aplikácie, ako ľahké stupnice, ale aj ťažšie piesne sú teda súčasťou aplikácie *The Piano Master*. MIDI súbory je možné nájsť na internete ako

OpenSource súbory¹. Je taktiež možné vytvárať MIDI súbory v rôznych či už desktopových alebo mobilných aplikáciách. Hudobník môže vytvárať MIDI súbor z piesne, ktorú hrá na hudobnom nástroji, ktorý je pripojený k príslušnému softwaru. Existuje teda široké spektrum možností ako sa dostať k MIDI súboru. V aplikácii *The Piano Master* je teda demonštrovaná jedna z možností využitia MIDI súborov.

Všetky tieto výukové lekcie majú dostupné všetci užívatelia aplikácie *The Piano Master*, avšak jednou z funkcií aplikácie je aj vytváranie lekcí užívateľmi v editačnom móde aplikácie. Cieľom aplikácie bolo, aby tieto novo vytvorené lekcie boli dostupné všetkým ostatným užívateľom. V aplikácii *The Piano Master* je vytvorená verejná iCloudová databáza (*private let database = CKContainer.default().publicCloudDatabase*), ktorá je prepojená s databázou, ktorá je priradená k aplikácii *The Piano Master* a dá sa spravovať v nástroji *CloudKit Dashboard*. V aplikácii sa využíva trieda *SongIC* a trieda *SongHardIC*. Tieto triedy a ich atribúty su bližšie popísané v kapitole 3.4 *Návrh komunikácie mobilnej aplikácie*. Pomocou funkcií *saveEasyIC()*, *saveHardIC()* a *addErrorand()* sa uložia potrebné objekty vytvorené po vzore tried *SongIC* a *SongHardIC* na iCloud do vytvorenej verejnej databázy.

Vytvorené lekcie sú teda uložené vo verejnej iCloudovej databáze. Teraz je však potrebné sprístupniť tieto lekcie všetkým ostatným užívateľom. Pomocou funkcie *updateErrands()*, ktorá je implementovaná v aplikácii *The Piano Master* sme schopní po pripojení na internet v reálnom čase zobrazíť obsah verejnej iCloud databázy korešpondujúcej k aplikácii. Obsah databázy sa zobrazuje do dvoch tabuliek *TableView*. Do jednej tabuľky sa zobrazujú lekcie, ktoré spadajú do ľahšej kategórie, a do druhej tabuľky sa zobrazia ťažšie lekcie.

Užívateľ má možnosť uložiť si niektoré lekcie do zoznamu obľúbených lekcí. V tomto prípade sa lekcía uloží pomocou implementovanej funkcie *save()*. Funkcia *save()* uloží obsah jednej lekcie z verejnej databázy iCloud do Core Data - persistentného úložiska na zariadení. Zoznam obľúbených lekcí je taktiež uložený v tabuľke *TableView* a je dostupný aj keď užívateľ nemá pripojenie k internetu. Užívateľ má možnosť vymazať lekcíu zo zoznamu obľúbených lekcí. Táto funkcionálna je zabezpečená v aplikácii *The Piano Master* pomocou funkcie *removeFavouriteItem()*. Implementáciu funkcie *removeFavouriteItem()* je možné vidieť pod týmto textom.

```
// Remove favourite item
func removeFavouriteItem(atIndexPath indexPath: IndexPath){

    guard let managedContext = appDelegate?.persistentContainer.viewContext
    else {
        return
    }

    managedContext.delete(favouriteSongItemsArray[indexPath.row])
    do {
        try managedContext.save()
        print("Succesfully removed Song!")
    } catch {
        debugPrint("Could not remove: \(error.localizedDescription)")
    }
}
```

¹<https://www.partnersinrhyme.com/midi/>

4.3 Generovanie zvuku

Aplikácia *The Piano Master* umožňuje zvukové prehrávanie jednotlivých lekcí a piesní, toto prehrávanie je zabezpečené pomocou *piano.sf2* súboru. Každá nota, ktorej popis je uložený v MIDI súbore je interpretovaná práve týmto *piano.sf2* súborom. V súbore *piano.sf2* sú uložené zvukové vzorky všetkých 88 kláves, potrebných na reprezentovanie zvuku klavíra.

Keď je teda napríklad v MIDI súbore zaznamenaná nota C3, ktorá je v štandarde MIDI označená hodnotou 60, je potrebné reprezentovať túto notu zvukom danej klávesy s hodnotou C3. V aplikácii *The Piano Master* bolo potrebné implementovať funkciu *playNote(note: UInt32)*, kde parameter *note* značí hodnotu noty z MIDI súboru konkrétnej piesne. Vo vyššie spomenutom prípade noty C3 bude parametrom tejto funkcie hodnota 60 (označenie noty C3 v MIDI súbore). Vo funkcii *playNote(note: UInt32)* je implementovaná funkcionálna na prehratie noty z parametru funkcie pomocou súboru *piano.sf2*. Súbor *piano.sf2* teda dostane požiadavku na zahratie práve tejto noty s hodnotou 60 z MIDI súboru. V MIDI súbore piesne je tiež popísané v akom okamihu má byť nota reprezentovaná a aké dlhé má byť trvanie noty. Z týchto hodnôt je vypočítané, aká dlhá je nota, aké má trvanie a teda kedy končí aj zvuková reprezentácia noty. Na ukončenie prehrávania nôt bola v aplikácii *The Piano Master* implementovaná funkcia *stopNote(note: UInt32)*, ktorej parameter *note* značí hodnotu noty z MIDI súboru, ktorej prehrávanie práve skončilo. Vo vyššie spomenutom príklade noty C3 bude parametrom funkcie *stopNote(note: UInt32)* hodnota 60. Implementáciu tejto funkcionality môžeme vidieť nižšie pod týmto textom.

Pod textom môžeme ďalej vidieť implementáciu funkcie *loadMIDISynthSoundFont()*, ktorá slúži na načítanie súboru *piano.sf2*.

```
// Vars
let soundFontFileName = "piano"
let soundFontFileExt = ".sf2"

// This function will load the default sound font and set the synth
unit's property.
func loadMIDISynthSoundFont() {

    if var bankURL = Bundle.main.url(
        forResource: soundFontFileName,
        withExtension: soundFontFileExt) {

        let status = AudioUnitSetProperty(
            self.midisynthUnit!,
            AudioUnitPropertyID(kMusicDeviceProperty_SoundBankURL),
            AudioUnitScope(kAudioUnitScope_Global),
            0,
            &bankURL,
            UInt32(MemoryLayout<URL>.size))
    } else {
        print("Could not load sound font")
    }
}
```

```

// Play a note as message using note param
func playNote(note: UInt32) {

    let channel = UInt32(0)
    let noteCommand = UInt32(0x90 | channel)
    var status = OSStatus(noErr)

    generatePitch()

    status = MusicDeviceMIDIEvent(
        self.midisynthUnit!,
        noteCommand,
        note,
        128,
        0)
}

// Stop playnig a note as message using note param
func stopNote(note: UInt32) {

    let channel = UInt32(0)
    let noteCommand = UInt32(0x80 | channel)
    var status = OSStatus(noErr)

    generatePitch()

    status = MusicDeviceMIDIEvent(
        self.midisynthUnit!,
        noteCommand,
        note,
        0,
        0)
}

```

Kapitola 5

Testovanie

V poslednej kapitole je popísané, akým spôsobom bola aplikácia *The Piano Master* testovaná. Je tu popísané testovanie na užívateľoch, spätná väzba od týchto užívateľov formou dotazníka, a taktiež formou diskusie po testovaní, vyhodnotenie získaných dát a možné vylepšenia a rozšírenia aplikácie.

5.1 Vstupné okolnosti testovania

Na testovanie aplikácie *The Piano Master* boli vybrané 3 osoby. Jeden zo zúčastnených mal pred testovaním základné skúsenosti s hrou na klávesové nástroje. Ďalší zo zúčastnených nemal pred testovaním žiadne skúsenosti. Posledná osoba mala dlhoročné skúsenosti s hrou na klávesové nástroje. Všetky zúčastnené testovacie osoby dostali rýchle zaškolenie a pokyny zoznámiť sa s aplikáciou. Ďalej si mali prejsť základné lekcie hry (stupnice) a následne mali vyskúšať ťažšiu lekcii, kde mali za úlohu zahrať niektorú z vybraných piesní. Testovacie osoby si mali vytvoriť svoj obľúbený zoznam piesní. Ďalej si mali vytvoriť krátku lekcii v editačnom móde aplikácie, a následne mali cez aplikáciu formou príspevku informovať komunitu užívateľov na sociálnej sieti Facebook o vytvorení lekcii. Nasledujúci postup popisuje zadané úlohy, ktoré užívatelia obdržali.

1. Spustíte aplikáciu.
2. Vyberte si ľubovoľné 3 lekcii z ponuky jednoduchších lekcii.
3. Vypočujte si každú z vybraných lekcii.
4. V každej z vybraných lekcii zvolte tempo na hodnotu 25 a pokúste sa lekcii zahrať.
5. V každej z vybraných lekcii zvolte tempo na hodnotu 60 a pokúste sa lekcii zahrať.
6. Následne zmeňte tempo na hodnotu 120 a pokúste sa zahrať lekcii s týmto výrazne rýchlejším tempom.
7. Z 3 vybraných lekcii si uložte 2 lekcii do sekcie „Favourite“.
8. Vyberte si ľubovoľnú jednu pieseň z ponuky ťažších lekcii.
9. Vypočujte si túto vybranú pieseň.
10. Zvolte si precvičovanie len pravej ruky a pokúste sa pieseň zahrať.

11. Zvoľte si precvičovanie len ľavej ruky a pokúste sa pieseň zahrať.
12. Zvoľte si precvičovanie oboch rúk, nastavte tempo piesne na nižšiu hodnotu a pokúste sa pieseň zahrať.
13. Nastavte tempo piesne na vyššiu hodnotu (cca. 120) a pokúste sa pieseň zahrať.
14. Vyberte si ťažší úsek piesne a cyklickým opakovaním sa pokúste tento úsek nacvičiť.
15. Uložte si túto pieseň do zoznamu obľúbených piesni.
16. V editačnej časti aplikácie vytvorte jednoduchú lekciu s využitím 15 nôt.
17. Doplníte dodatočné informácie o lekcií ako názov, kategóriu, atď. Lekciu uložte.
18. Vyhľadajte vytvorenú lekciu v zozname všetkých zdieľaných, užívateľmi vytvorených lekcii.
19. Uložte si túto lekciu do obľúbených lekcii, aby ste si ju mohli prehrať aj v prípade, keď zariadenie nebude mať prístup k internetovému pripojeniu.
20. Prehrajte si vytvorenú lekciu v prezentačnom móde aplikácie.
21. Informujte pomocou aplikácie *The Piano Master* Vašich kamarátov na sociálnej sieti Facebook o vytvorení Vašej lekcie. Pridajte popis k vygenerovanému príspevku.
22. Vypnite aplikáciu *The Piano Master*.

Súčasťou testu bol aj dotazník s otázkami zameranými na pohodlie pri práci s aplikáciou, mierou kontroly nad aplikáciou a hodnotením náročnosti zvolených úloh. Tieto otázky sú uvedené v Tabuľke 5.1. Tento dotazník je analyzovaný v kapitole 5.2. Následne prebehla diskusia s testovacími osobami. Zhrnutie diskusie si je možné pozrieť v kapitole 5.3.

1.	Ako dlho Vám trvalo splnenie všetkých zadaných úloh?
2.	Využívali by ste častejšie prezentačný alebo editačný mód aplikácie?
3.	Využívali by ste možnosť komunitného zdieľania lekcii na sociálnych sieťach typu Facebook?
4.	Ako hodnotíte prehľadnosť systému? (známka 1 – 10)
4.	Odporučili by ste aplikáciu <i>The Piano Master</i> ďalším užívateľom?

Tabuľka 5.1: Otázky použité v dotazníku pre testovanie aplikácie *The Piano Master*.

5.2 Analýza dotazníku

V Tabuľke 5.2 je možné vidieť vyplnený dotazník od troch testovacích osôb.

Pri bližšej analýze vyplneného dotazníka je možné vidieť, že zadané úlohy zabrali testovacím užívateľom pomerne dlhší čas. Je možné vidieť, že skúsenému klavíristovi trvalo splnenie zadaných úloh menšie množstvo času ako testovacej osobe bez skúsenosti s hrou na klávesové nástroje.

Z tabuľky je ďalej možné vyčítať, že aplikáciu *The Piano Master* by väčšina užívateľov používala na precvičovanie lekcii v prezentačnom móde aplikácie.

Všetky tri zúčastnené testovacie osoby by aplikáciu *The Piano Master* využívali aj na komunitné zdieľanie vytvorených lekcí na sociálnych sieťach.

Prehľadnosť systému bola užívateľmi ohodnotená v škále 1 - 10 hodnotami 4 a 2-krát 2 čo je nadpriemerný výsledok, avšak stále tu je priestor na zlepšovanie.

Všetky tri zúčastnené testovacie osoby by aplikáciu *The Piano Master* odporučili svojim známym.

Otázka	Užívateľ bez skúsenosti s hrou na klavíri.	Užívateľ - začiatočník v hre na klavíri.	Užívateľ s dlhoročnými skúsenosťami s hrou na klavíri.
1.	2 hodiny	90 minút	70 minút
2.	prezentačný	editačný	prezentačný
3.	určíte áno	áno	áno
4.	4	2	2
5.	áno	áno	áno

Tabulka 5.2: Výsledky dotazníka.

5.3 Diskusia s užívateľmi

Po testovaní aplikácie jednotlivými užívateľmi a po vyplnení dotazníka prebehla krátka diskusia, ako vnímali aplikáciu *The Piano Master*.

Užívateľ bez skúsenosti s hrou na klavíri

Testovacia osoba, ktorá nemala žiadne skúsenosti s hrou na klávesové nástroje zkonštatovala, že by uvítala väčšie množstvo základných cvičení. Ďalej podotkla, že pri dlhodobom používaní aplikácie *The Piano Master* by jej pravdepodobne prekážala absencia užívateľmi vytvorených lekcí v prípade, že by nemala prístup na internet. Avšak ako výhodu videla ukladanie si obľúbených lekcí do zariadenia a teda možnosť použiť tieto lekcie aj v prípade, že momentálne nebude pripojená k internetu. Ďalej poznamenala, že ovládacie prvky v editačnom móde aplikácie by mohli byť v prípadnej druhej verzii väčšie.

Užívateľ - začiatočník v hre na klavíri

Testovacia osoba, ktorá mala základné skúsenosti s hrou na klávesové nástroje zkonštatovala, že by určite aplikáciu *The Piano Master* pravidelne využívala na tvorbu lekcí, a tým aj umožnila ostatným užívateľom poskytnúť svoje vytvorené lekcie. Taktiež sa jej páčila možnosť zdieľania lekcí medzi komunitou užívateľov na sociálnej sieti. Testovacia osoba videla túto možnosť za vhodný spôsob ako propagovať svoje vytvorené lekcie aj medzi užívateľmi, ktorí zatiaľ nemajú nainštalovanú aplikáciu *The Piano Master*.

Užívateľ s dlhoročnými skúsenosťami s hrou na klavíri

Tretia testovacia osoba, profesionál, v hre na klávesové nástroje zkonštatoval, že by aplikáciu určite používal a to hlavne na precvičovanie ťažších piesní. Následne poznamenal, že pri dlhodobom používaní aplikácie *The Piano Master* by uvítal väčšie množstvo náročnejších piesní. Taktiež spomenul, že by si vedel predstaviť pravidelné aktualizácie, kde by sa

aktualizoval zoznam zložitejších piesní o nové lekcie. Tento užívateľ by si tiež vedel predstaviť prepojiť aplikáciu *The Piano Master* s digitálnym klavírom s možnosťou vzájomnej komunikácie týchto dvoch systémov.

5.4 Odhalené nedostatky

Z vyplneného dotazníku, a z diskusie, ktorá prebehla po testovaní užívateľmi sú v tejto kapitole zhrnuté odhalené nedostatky aplikácie *The Piano Master*.

- Absencie väčšieho množstva ľahších lekcií.
- Absencia väčšieho množstva ťažších lekcií.
- Absencia využívania užívateľmi vytvorených lekcií v prípade, keď zariadenie nie je pripojené k internetu.
- Malé ovládacie prvky v editačnom móde aplikácie.
- Absencia prepojenie aplikácie *The Piano Master* s klávesovým nástrojom s možnosťou komunikácie týchto dvoch systémov.

5.5 Zhrnutie testovania

Testovanie odhalilo niektoré pozitívne, ale aj negatívne aspekty aplikácie *The Piano Master*. Testovaním sa zistilo, že v druhej verzii aplikácie *The Piano Master* by bolo vhodné pridať väčšieho množstva lekcií. Ďalej by bolo vhodné zamyslieť sa nad ovládacími prvkami v editačnom móde aplikácie *The Piano Master* s následnou vhodnou úpravou veľkosti ovládacích prvkov. Aplikáciu *The Piano Master* by vylepšilo aj prepojenie s digitálnymi klávesovými nástrojmi pre lepšiu odozvu pri hre užívateľa na klávesovom nástroji.

Kapitola 6

Záver

Cieľom tejto práce bolo vytvoriť mobilnú aplikáciu pre podporu výučby hry na klávesové nástroje. Výsledkom je mobilná aplikácia *The Piano Master* pre mobilné zariadenia s operačným systémom iOS. Aplikácia je optimalizovaná ako na zariadenia iPhone tak aj na zariadenia typu iPad. Aplikácia *The Piano Master* bola implementovaná v programovacom jazyku Swift a spĺňa všetky požiadavky zadania. Aplikácia *The Piano Master* umožňuje užívateľovi cvičiť hru na klavíri pomocou jednoduchších, ale aj ťažších lekcí. Taktiež umožňuje cvičiť lekcie vytvorené inými užívateľmi aplikácie. Aplikácia *The Piano Master* umožňuje užívateľskú tvorbu lekcí hry a zdieľanie vytvorených lekcí medzi komunitou užívateľov. V aplikácii *The Piano Master* bola implementovaná funkcionálna jednotlivých aspektov dobrej formy výučby hry na klávesové nástroje. A teda aplikácia *The Piano Master* umožňuje výučbu stupníc a taktiež aj výučbu rôznych hudobných žánrov a to formou jednotlivých skladieb. V aplikácii *The Piano Master* je možné meniť tempo jednotlivých jednoduchších lekcí a taktiež aj zložitejších skladieb ako aj všetkých ostatných lekcí ktoré zdieľali ostatní užívatelia. Ďalej ma užívateľ možnosť výučby hry každej ruky zvlášť, ale aj výučby hry oboch rúk naraz. Aplikácia *The Piano Master* taktiež umožňuje výučbu štvorročných skladieb (skladby stavané pre hru dvoch ľudí na jednom klávesovom nástroji). Aplikácia *The Piano Master* poskytuje možnosť označenia jednotlivých kláves na klaviatúre, ďalej umožňuje označenie hodnôt nôt v jednotlivých lekciami. Užívateľovi je umožnené aj cyklické opakovanie jednotlivých pasáží v skladbe.

Aplikácia *The Piano Master* je plne funkčná, no napriek tomu sa počas testovania aplikácie ukázalo, že by bolo vhodné niektoré prvky aplikácie *The Piano Master* mierne upraviť, prípadne dodať určité prvky a funkcionálnu, ktoré by zvýšili mieru jej konkurencieschopnosti.

Vhodným vylepšením aplikácie *The Piano Master* je pridanie väčšieho množstva lekcí. Ďalej by bolo vhodné zamyslieť sa nad ovládacími prvkami v editačnom móde aplikácie *The Piano Master*, s následnou vhodnou úpravou rozloženia a veľkosti ovládacích prvkov.

Možným vylepšením aplikácie *The Piano Master* by mohla byť vzájomná komunikácia s digitálnymi klávesovými nástrojmi, ktoré v sebe majú integrované MIDI rozhranie. Toto rozšírenie by nemuselo byť až tak náročné, pretože aplikácia *The Piano Master* pracuje s formátom MIDI. Toto prepojenie by mohlo byť využité napríklad v prípade, že by užívateľ hral skladbu na klávesovom nástroji, a skladba by sa mu ukladala priamo do aplikácie *The Piano Master* vo forme MIDI súborov, následne by mohol túto skladbu pomocou aplikácie *The Piano Master* zdieľať s ostatnými užívateľmi. Ďalším možným vylepšením v prípade prepojenia aplikácie *The Piano Master* s digitálnymi klávesovými nástrojmi, by mohla byť kontrola, či užívateľ zahral správne lekcie, ktoré mu aplikácia *The Piano Master* ponúka.

Napríklad pokiaľ by užívateľ zahral v určitý čas nesprávnu notu, aplikácia *The Piano Master* by ho na to upozornila, a mohla by mu automaticky ponúknuť cyklické opakovanie daného zložitejšieho úseku. Vývoj aplikácie *The Piano Master* nekončí a plánujem v budúcnosti tieto vylepšenia realizovať.

Literatura

- [1] *Apple Inc.: About Instruments*. [Online; navštívené 20.04.2019].
URL <https://help.apple.com/instruments/mac/current/#/dev7b09c84f5>
- [2] *Apple Inc.: CloudKit*. [Online; navštívené 06.04.2019].
URL <https://developer.apple.com/documentation/cloudkit>
- [3] *Apple Inc.: iCloud*. [Online; navštívené 07.04.2019].
URL <https://developer.apple.com/icloud/>
- [4] *Apple Inc.: Initializing the Core Data Stack*. [Online; navštívené 09.04.2019].
URL https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CoreData/InitializingtheCoreDataStack.html#//apple_ref/doc/uid/TP40001075-CH4-SW1
- [5] *Apple Inc.: Understanding Auto Layout*. [Online; navštívené 22.04.2019].
URL https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/AutolayoutPG/index.html#//apple_ref/doc/uid/TP40010853-CH7-SW1
- [6] *Apple Inc.: What Is Core Data?* [Online; navštívené 06.04.2019].
URL https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CoreData/index.html#//apple_ref/doc/uid/TP40001075-CH2-SW1
- [7] *MIDI Manufacturers Association: Summary of MIDI Messages*. [Online; navštívené 15.04.2019].
URL <http://www.midi.org/techspecs/midimessages.php>
- [8] *MIDI Manufacturers Association: What is MIDI?* [Online; navštívené 12.04.2019].
URL <https://www.midi.org/>
- [9] *Soundfont*. [Online; navštívené 14.04.2019].
URL <https://cs.wikiversity.org/wiki/Varhany/Soundfont>
- [10] *SoundFont*. [Online; navštívené 20.04.2019].
URL <https://en.wikipedia.org/wiki/SoundFont>
- [11] *SoundFont® Technical Specification*. [Online; navštívené 08.04.2019].
URL <http://www.synthfont.com/sfspec24.pdf>
- [12] Beňak, P.: *Co je to SoundFont?* [Online; navštívené 13.04.2019].
URL <http://mag.repro.cz/view.php?cisloclanku=2003121002>

- [13] Chang, C. C.: *Fundamentals of Piano Practice* . Booksurge, 2009, ISBN 1-4196-7859-0.
- [14] Grant, C.: *iOS 9 Day by Day*. Scott Logic Ltd, 2014 - 2016, ISBN 9781326532055.
- [15] Jirsák, M.: *Seri(ál)ová MIDI komunikace*. [Online; navštívené 10.04.2019].
URL <http://www.muzikus.cz/tagy/serialova-midi-komunikace/>
- [16] Luboslav Lacko: *Vývoj aplikací pro iOS* . Albatros Media a.s., 2018, ISBN 978-80-251-4942-3.
- [17] Neuburg, M.: *Programming iOS 9: Dive Deep into Views, View Controllers, and Frameworks*. . O'Reilly Media, December 2015, ISBN 978-1-491-93685-6.
- [18] Smyth, N.: *iOS 9 App Development Essentials – First Edition* . CreateSpace Independent Publishing Platform; 1 edition, 2015, ISBN 978-1518651557.