



Bakalářská práce

Stanice pro lekce hudební nauky

Studijní program:

B0613A140005 Informační technologie

Studijní obor:

Aplikovaná informatika

Autor práce:

Jakub Kindermann

Vedoucí práce:

Ing. Jana Kolaja Ehlerová, Ph.D.

Ústav nových technologií a aplikované
informatiky

Liberec 2024



Zadání bakalářské práce

Stanice pro lekce hudební nauky

<i>Jméno a příjmení:</i>	Jakub Kindermann
<i>Osobní číslo:</i>	M20000037
<i>Studijní program:</i>	B0613A140005 Informační technologie
<i>Specializace:</i>	Aplikovaná informatika
<i>Zadávající katedra:</i>	Ústav nových technologií a aplikované informatiky
<i>Akademický rok:</i>	2022/2023

Zásady pro vypracování:

1. Proveďte rešerši aplikací k výuce hudební teorie.
2. Navrhněte stanici založenou na Raspberry Pi s fyzickou klaviaturou a potřebnými periferiemi. Nainstalujte a otestujte ji.
3. Navrhněte formát lekcí a cvičení, zahrňte prvky gamifikace, možnost exportu i importu lekcí a propojení s webovou aplikací. Návrh implementujte.
4. Otestujte řešení a navrhněte možná vylepšení.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: 30 – 40 stran
Forma zpracování práce: tištěná/elektronická
Jazyk práce: čeština

Seznam odborné literatury:

- [1] MONK, Simon. Raspberry Pi cookbook. Second edition. Beijing: O'Reilly, 2016. ISBN 978-1-491-93910-9.
- [2] REŽNÝ, Pavel. Elementární hudební teorie I. 2., nezměněné vydání. Olomouc: Univerzita Palackého v Olomouci, Pedagogická fakulta, 2018. Studijní literatura. ISBN 978-80-244-5422-1.
- [3] REŽNÝ, Pavel. Elementární hudební teorie II. 2., nezměněné vydání. Olomouc: Univerzita Palackého v Olomouci, Pedagogická fakulta, 2018. Studijní literatura. ISBN 978-80-244-5423-8.
- [4] SYROVÝ, Václav. Hudební zvuk: příspěvek k teorii zvukové tvorby. V Praze: Akademie múzických umění, 2009. Akustická knihovna Zvukového studia Hudební fakulty AMU, 9. ISBN 978-80-7331-161-2.

Vedoucí práce: Ing. Jana Kolaja Ehlerová, Ph.D.
Ústav nových technologií a aplikované informatiky

Datum zadání práce: 12. října 2022
Předpokládaný termín odevzdání: 15. května 2024

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Josef Chaloupka, Ph.D.
garant studijního programu

V Liberci dne 19. října 2022

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

Stanice pro lekce hudební nauky

Abstrakt

Tato práce zkoumá inovativní přístup k výuce hudby a hudební teorie spojením tradičních metod s interaktivními a zábavnými prvky. Provedl jsem rešerši existujících vzdělávacích aplikací, nastudoval historii klavíru a výuky hudby a zkoumal principy gamifikace. Následně navrhl a sestavil vzdělávací stanici, která je založena na Raspberry Pi, s fyzickou klaviaturou a dotykovým displejem propojenou s webovou aplikací vytvořenou pomocí frameworku Django. Cílem bylo vytvořit platformu s možností dalšího rozvoje a přizpůsobení se novým potřebám a technologickým trendům. Práce obsahuje popis použitého hardware, návrh a implementaci aplikace pro stanici a webové aplikace. Nakonec jsem provedl testování a navrhl možná vylepšení pro budoucí vývoj. Tento projekt představuje snahu o obohacení výuky hudby prostřednictvím digitální technologie.

Klíčová slova: Raspberry Pi, Django, piano, hudební nauka, gamifikace

Station for music theory lessons

Abstract

This thesis explores an innovative approach to teaching music and music theory by combining traditional methods with interactive and entertaining elements. I have conducted research on existing educational applications, studied the history of piano and music education, and explored the principles of gamification. Then I designed and built an educational station based on a Raspberry Pi computer with a physical keyboard and touchscreen connected to a web application created using the Django framework. The aim was to create a platform with the possibility of further development and adaptation to new needs and technological trends. The work includes a description of the hardware used, the design and implementation of the station application and the web application. Finally, I performed testing and suggested possible improvements for future development. This project represents an effort to enrich music education through digital technology.

Keywords: Raspberry Pi, Django, piano, music learning, gamification

Poděkování

Děkuji Ing. Janě Kolaje Ehlerové Ph.D., vedoucí mé bakalářské práce za podnětné konzultace a rady. Za poskytnutí 3D tiskárny děkuji svému strýci Josefovi Krennovi. Nakonec děkuji své přítelkyni Sylvii Kadlec a kamarádovi Albertovi Pospíšilovi za korekturu.

Obsah

Seznam zkratk	11
1 Úvod	12
2 Rešerše	13
2.1 Počátky klavíru a jeho výuky	13
2.1.1 Počátky klavíru	13
2.1.2 Počátky výuky klavíru a varhan	13
2.2 Gamifikace a učení pomocí aplikací ve vzdělávání	13
2.3 Související práce	14
2.3.1 GUIDO system	14
2.3.2 Piano Tutor	15
2.3.3 Musictheory.net	15
2.3.4 Yousician	16
2.3.5 Duolingo music	17
3 Hardware, sestavení a zprovoznění stanice	18
3.1 Použitý hardware	18
3.1.1 Raspberry Pi 3 Model B	18
3.1.2 Piano HAT	19
3.1.3 Adafruit PiTFT - 320x240 2.8 TFT+Touchscreen	20
3.2 Sestavení komponent	20
3.3 Návrh a vytisknutí pouzdra	20
4 Návrh a implementace aplikace pro stanici	23
4.1 Operační systém a ovladače	23
4.2 Funkční požadavky	23
4.3 Návrh aplikace	24
4.3.1 Výběr knihoven a programů pro vytvoření aplikace	24
4.3.2 Databáze	24
4.4 Implementace aplikace	25
4.4.1 Implementace vlastních objektů grafického rozhraní	26
4.4.2 Dotyková klávesnice a klaviatura	26
4.4.3 API a databáze	26
4.4.4 Grafické rozhraní	27
4.4.5 Generování notového zápisu pomocí Lilypond	28

4.4.6	Syntetizace zvuku pomocí FluidSynth	29
5	Návrh a řešení webové aplikace	30
5.1	Funkční požadavky	30
5.1.1	Modul učitele	30
5.1.2	Modul API	30
5.1.3	Modul administrátora	31
5.2	Ostatní požadavky	31
5.3	Návrh aplikace	31
5.4	Výběr knihoven	31
5.5	Výběr databáze	32
5.6	Implementace aplikace	32
5.6.1	Django konvence	32
5.6.2	Modul API	32
5.6.3	Modul učitele	33
5.6.4	Grafické rozhraní modulu učitele	35
5.7	Nasazení a zabezpečení	35
5.7.1	Nasazení	36
6	Testování a návrhy na zlepšení	38
6.1	Testování	38
6.2	Testování z pohledu uživatele	38
6.3	Návrhy na zlepšení	39
6.3.1	Rozšíření lekcí a přidání hudební teorie	39
6.3.2	Další prvky gamifikace	39
6.3.3	Využití vlastní desky	39
7	Závěr	40
A	Přílohy	43
A.1	Zdrojové kódy	43
A.2	Sestavení komponent - schéma	43

Seznam obrázků

2.1	Piano tutor (8)	15
2.2	Podoba lekce v aplikaci Yousician	16
3.1	Raspberry 3 V1.2 (17)	19
3.2	Piano HAT	19
3.3	Adafruit PiTFT	20
3.4	Sestavení komponent - vizualizace. (poz. Nepájivé pole je zde pouze pro zjednodušení vizualizace. V zařízení není.)	21
3.5	Design pouzdra v programu Onshape	22
3.6	Výsledná podoba sestavení	22
4.1	Diagram webové a lokální aplikace	25
4.2	Databázový diagram	27
4.3	Snímky grafického rozhraní	28
5.1	Snímky grafického rozhraní modulu API	33
5.2	Databázové schéma webové aplikace	34
5.3	Snímky grafického rozhraní webové aplikace	36

Seznam zkratek

HAT	Hardware attached on top
GB	Gigabyte
GPIO	General-purpose input/output
GHz	Gigahertz
USB	Universal Serial Bus
I2C	Inter-Integrated Circuit
PWM	Pulse-width modulation
EEPROM	Electrically erasable programmable read-only memory
PiTFT	Raspberry Pi thin film transistor display
CAD	Computer-aided design
3D	Three-dimensional
PLA	Polylactic Acid
API	Application Programming Interface
MIDI	Musical Instrument Digital Interface
SQL	Structured query language
REST	Representational state transfer
ID	Identifier
XP	Experience point
JSON	JavaScript Object Notation
CSS	Cascading Style Sheets
JS	JavaScript
ORM	Object–relational mapping
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
MITM	Man-in-the-middle
SSH	Secure Shell
LLM	Large language mode

1 Úvod

V digitální éře, kde technologie ovlivňuje téměř každý aspekt života, je nevyhnutelné přizpůsobit vzdělávání a učení se novým dovednostem rychle se měnícímu digitálnímu prostředí. Zvláště v oblasti hudby a hudební teorie, kde se tradiční metody výuky střetávají s mou touhou po interaktivitě a zábavnosti, jsem viděl prostor pro inovativní přístup.

Ve své práci jsem provedl rešerši existujících aplikací pro výuku hudební teorie, prostudoval jsem historii klavíru a výuky hudby a zkoumal jsem principy gamifikace a využití aplikací ve vzdělávání. Poté jsem navrhl a sestavil vlastní vzdělávací stanici s fyzickou klaviaturou a dotykovým displejem, která byla schopna propojení s webovou aplikací pro další rozšíření možností výuky.

Mým záměrem nebylo vytvořit jednorázový produkt, ale spíše vybudovat platformu, která bude schopna se neustále rozvíjet a přizpůsobovat novým potřebám a technologickým trendům. V závěrečné části této práce jsem provedl testování svého řešení a navrhl jsem možná vylepšení pro budoucí vývoj.

S touto ambicí jsem se ponořil do detailů navrhování, implementace a testování své moderní vzdělávací stanice pro hudební teorii a praxi. Cílem zároveň není nahradit tradiční výuku, ale spíše ji doplnit a obohatit novými možnostmi a způsoby učení. Věřím, že digitální technologie může být silným nástrojem pro vzdělávání a že je na nás, abychom ji využili k dosažení nových úrovní znalostí a dovedností.

2 Rešerše

2.1 Počátky klavíru a jeho výuky

Podle Slovníku spisovné češtiny se jedná o „*strunový nástroj s kladívkovou mechanikou uváděnou do pohybu klávesami*“ (1). Jedná se tedy o jeden z tzv. klávesový nástrojů, tj. skupinu nástrojů, na které se hraje pomocí klaviatury (soustavy kláves). Do stejné kategorie řadíme například varhany, syntentizátory a cembala. Podle Sachsovy–Hornbostelovy klasifikace hudebních nástrojů se jedná o takzvaný chordofon, tedy strunný nástroj, jehož zvuk vzniká kmitáním struny (2, s. 8). Chordofony jsou široká skupina nástrojů a tato klasifikace tedy do jediné kategorie řadí klavír, housle i harfu.

2.1.1 Počátky klavíru

Historie hudby a historie klavíru se vinou neoddělitelně spolu, vzájemně se ovlivňují a formují. Od prvních nástrojů vyrobených člověkem a zvuků vytvářených přírodními elementy po složité orchestrální skladby a virtuózní sóla, vývoj hudby byl neustálým průvodcem lidské kultury a vyjádřením lidských emocí a myšlenek.

Klavír, jakožto jeden z nejvýznamnějších hudebních nástrojů současnosti, má hluboké kořeny v minulosti. Jeho předchůdce, cembalo, hrálo klíčovou roli ve vývoji klaviaturní technologie. Avšak průlom přišel v 18. století, kdy italský nástrojař Bartolomeo Cristofori vyvinul první pianoforte (2, s. 32), což byl předchůdce moderního klavíru. Od té doby se klavír stal středobodem klasické hudby, jazzu a mnoha dalších hudebních žánrů.

2.1.2 Počátky výuky klavíru a varhan

Podle Václava Sýkory je nejstarší klávesovou a varhanní učebnicí *Fundamentum organiscandi* od Conrada Paumanna z roku 1452. Jeho učebnice však není výkladem či návodem, ale praktickými příklady pro improvizaci (2, s. 63). Za skutečný rozvoj klavírní výuky můžeme považovat polovinu 18. století, kdy Carl Philipp Emanuel Bach položil základ škol hry na klavír svým vydáním *Versuch über die wahre Art das Clavier zu spielen (Studie o pravém umění klavírní hry)*. (2, s. 84)

2.2 Gamifikace a učení pomocí aplikací ve vzdělávání

Karl M. Kapp definuje gamifikaci ve své knize *The Gamification of Learning and Instruction* takto: „*Gamifikace je využití herních mechanik, estetiky a herního myšlení k za-*

pojení lidí, motivaci k činnosti, podpoře učení a řešení problémů“ (3, s. 10). Tato metoda tedy zahrnuje využití herních mechanik k posílení vnitřní motivace. Hlavním cílem je zlepšit motivaci a zapojení uživatelů do neherních úkolů.(4)

Gamifikace nabízí přístup ke vzdělávací praxi zaměřující se na kognitivní, emocionální a sociální aspekty učení. Podle studie z roku 2011(4, s. 3-4) může gamifikace ve výuce podpořit následující oblasti:

1. **Kognitivní:** Hry poskytují prostředí pro aktivní učení se prostřednictvím experimentování a řešení problémů. Udržují studenty motivované a angažované, a to prostřednictvím výzev odpovídajících úrovni dovedností hráče a nabízením vícero cest k úspěchu. Tento přístup může změnit postoj studentů k učení tím, že poskytuje jasné, akční úkoly a okamžité odměny, podporující snahu o neustálém zlepšování.(4, s. 3)
2. **Emoční:** Hry vyvolávají širokou škálu emocí, od zvědavosti přes frustraci až po radost. Důležité je, že pomáhají hráčům přijmout pocit selhání vytvořením bezpečného prostředí pro učení a rychlými zpětnými vazbami na provedenou práci. Tím, že selhání v bezpečném prostředí je nedílnou součástí procesu učení, gamifikace podporuje emoční odolnost a povzbuzuje studenty, aby viděli neúspěchy jako příležitosti k růstu, nikoli jako zdroje úzkosti.(4, s. 3)
3. **Sociální:** Hry umožňují hráčům zkoumat nové identity a role, jak fiktivní, tak osobní. Ve vzdělávacích prostředích může gamifikace poskytnout platformu pro přijetí identity učence, podporující pocit příslušnosti a zapojení do učení. Veřejným uznáním akademických úspěchů v rámci herního prostředí mohou studenti získat sociální důvěryhodnost a validaci svých úsilí, což posiluje jejich pozitivní školní identitu.(4, s. 4)

Dobře navržený systém gamifikace může využít tyto kognitivní, emocionální a sociální prvky k vytvoření transformačního učebního zážitku. Tím, že učení činí angažujícím, odměňujícím a sociálně významným, má gamifikace potenciál motivovat studenty k dosažení jejich potenciálu a zároveň transformovat jejich vnímání vzdělávání.

2.3 Související práce

Na cestě k interaktivní výuce hudební nauky a hry na piano byly potřeba také dva významné vynálezy a jejich rozvoj: osobní počítač a syntezátor. Oba se úspěšně rozšířily v sedmdesátých a osmdesátých letech dvacátého století. Existuje řada řešení určených k výuce hudebních dovedností. Mezi první systémy patří série interaktivních programů pro výuku hudební nauky od Hofstettera nazvaná GUIDO System.

2.3.1 GUIDO system

GUIDO system je aplikace pro výuku hudby, která byla vyvíjena od roku 1974 na University of Delaware. Aplikace předkládá studentům cvičení, jako je identifikace melodií nebo

notace rytmů. Dále umožňuje instruktorům přizpůsobit systém potřebám jednotlivých studentů prostřednictvím sady parametrů, které ovlivňují složitost generovaných hudebních cvičení (5).

2.3.2 Piano Tutor

V roce 1990 vyvinul Roger Dannenberg ve spolupráci s učiteli hry na klavír počítačovou aplikaci určenou k výuce hry na piano nazvanou Piano Tutor (6). Díky této aplikaci bylo možné získávat MIDI data z klavíru, která byla dále zpracovávána systémem. Aplikace obsahovala sadu cvičení ve formě hudebních not, které student musel hrát v rytmu metronomu. Systém automaticky otáčel stránky a byl schopen detekovat případné chyby v interpretaci studenta. Nakonec poskytoval zpětnou vazbu a navrhoval způsoby, jak se zlepšit. Aplikace byla vyvinuta v jazyce Common Lisp a C pro počítač Apple Macintosh a přinesla několik inovací (7):

1. Autoři implementovali software pro rozpoznání hudby, aby mohli sledovat postup studenta ve skladbě.
2. Další inovací je využití řady vstupních a výstupních zařízení, která rozšiřují aplikaci tak, aby byla atraktivní pro studenty. Mezi tato zařízení patřil akustický nebo elektronický klavír, přehrávač videodisků pro videozáznamy, počítačová grafika a hudba, a to včetně notového zápisu a syntézy hudby a řeči.
3. Třetí inovací je integrace techniky tvorby výukových materiálů do aplikace.



Obrázek 2.1: Piano tutor (8)

2.3.3 Musictheory.net

Musictheory.net je online platforma poskytující zdarma nástroje pro výuku hudby. Byla založena v roce 2000. Ricci Adams, zakladatel Musictheory.net, ji vytvořil jako zdroj pro výuku hudby. V roce 2011 opustil svou práci v Apple, aby se jí mohl plně věnovat. V té době rozšířil svou nabídku o aplikace Tenuto a Theory Lessons pro iOS (9).

Musictheory.net nabízí širokou škálu výukových materiálů a nástrojů pro studenty a učitele hudby. Mezi tyto zdroje patří interaktivní lekce teorie hudby, cvičení na rozpoznávání not, akordů a intervalů, stejně jako nástroje pro učení se čtení not (10).

Platforma umožňuje studentům individuálně postupovat v jejich vzdělávání a nabízí možnost opakování materiálu, pokud je to potřeba. Díky své online dostupnosti mohou studenti a učitelé přistupovat k materiálům kdykoli a odkudkoli, což usnadňuje učení a výuku hudby. Zároveň umožňuje učitelům vytvářet vlastní lekce, které mohou s žáky sdílet pomocí odkazu.

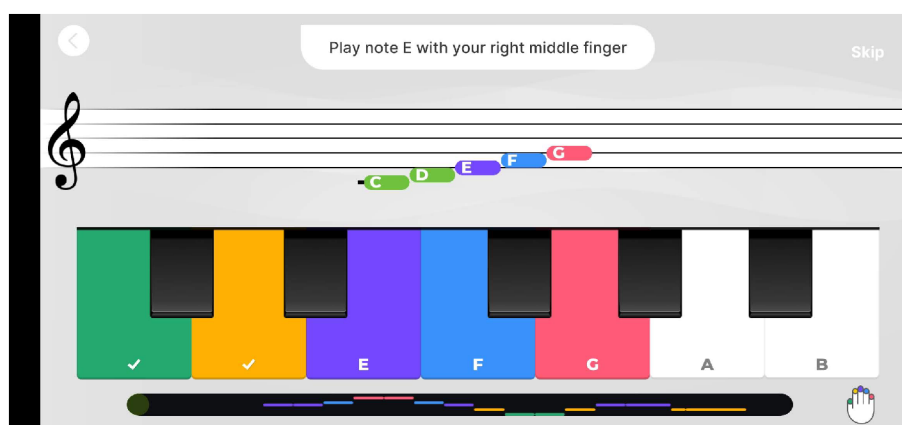
2.3.4 Yousician

Yousician je digitální platforma pro výuku hudby. Společnost Yousician Ltd. sídlí v Helsinkách a byla založena Chrisem Thürem a Mikkem Kaipainenem v roce 2010 pod názvem Ovelin. Prvním nástrojem, který Yousician dal k dispozici při svém uvedení na trh v roce 2014, byla kytara.(11)

Tato aplikace poskytuje okamžitou zpětnou vazbu na přesnost časování při odposlechu hudebních nástrojů pomocí mobilního zařízení nebo stolního počítače. Zároveň nabízí možnost využívat v rámci výuky pianu virtuálních kláves na displeji telefonu. V současnosti aplikace nabízí výuku čtyř hudebních nástrojů: kytary, klavíru, kontrabasů a ukulele, a mimo to také zpěvu.

Yousician přizpůsobuje lekce a cvičení na základě výkonu uživatelů a nabízí širokou škálu písní, cvičení, lekcí a tréninků. Uživatelé mohou sledovat svůj pokrok a monitorovat ho. Aplikace dále využívá prvky gamifikace k výuce hudby a poskytuje týdenní výzvy, které umožňují uživatelům soutěžit s ostatními na žebříčku. Obsah lekcí je strukturován a rozdělen do tří úrovní, přičemž dokončení jedné otevírá další. Přihlášení přes Facebook umožňuje uživatelům vyzvat své přátele k soutěži o lepší skóre (12).

Yousician není určen pouze pro studenty hudby, ale i pro učitele, kteří mohou využívat aplikaci jako výukový nástroj pro ukázky v hodinách. Aplikace rovněž obsahuje funkce, které nevyžadují fyzický hudební nástroj, jako jsou cvičení teorie hudby a poslech. Uživatelé mohou získat přístup pomocí měsíčního nebo ročního předplatného.



Obrázek 2.2: Podoba lekce v aplikaci Yousician

2.3.5 Duolingo music

Duolingo je aplikace původně vyvinuta pro výuku jazyků. Spuštěna byla oficiálně 19. června 2012. V roce 2024 měla nabídku 39 jazykových kurzů pro anglické mluvčí a zároveň kurz matematiky a hudební nauky. Ten společnost spustila v roce 2023 pro zařízení s operačním systémem iOS (13). Cvičení v rámci Duolingo Music zahrnuje tyto interaktivní úkoly:

- Rytmus
- Názvy not a jejich poloha na klavíru
- Jak se řadí noty za sebou
- Jak číst noty
- Trénink rozlišování not podle sluchu

3 Hardware, sestavení a zprovoznění stanice

Pro sestavení přenosné stanice jsem použil jako základní stavební kámen Raspberry Pi 3 Model B a dva rozšiřovací moduly (takzvané HAT). Protože při designu autoři níže zmíněných HATů nepočítali s potřebou připojení více zařízení, musel jsem tento problém vyřešit propojením pinů pomocí nepájivých kabelů a vytisknutím mnou navrženého pouzdra na 3D tiskárně, aby bylo možné stanici pohodlně ovládat.

3.1 Použitý hardware

3.1.1 Raspberry Pi 3 Model B

Od svého uvedení na trh v roce 2011 slouží Raspberry Pi jako cenově dostupný embedded počítač. Řada těchto počítačů si získala oblibu ve vědě, v průmyslu a mezi amatérskými uživateli, a to díky své jednoduchosti. V průběhu let se prodaly miliony kusů (14). Raspberry Pi 3 Model B obsahuje čtyřjádrový 64bitový procesor Broadcom BCM2837 s frekvencí 1,2 GHz a 1 GB operační paměti. Dále disponuje integrovaným BCM43438 čipem pro Wi-Fi a Bluetooth, ethernetovým portem, 40pinovým GPIO pro periferie a 4 USB 2.0 porty pro externí zařízení (15). Protože pro grafické rozhraní i další úkony spojené s během aplikace na stanici jsem zvolil retro stylizaci, která není náročná na zobrazování, je výkon tohoto počítače dostatečný.

GPIO piny lze dle dokumentace používat s celou řadou dalších funkcí, modul Piano HAT využívá ke komunikaci protokol I2C a modul displaye používá protokol SPI (16):

- PWM
 - Softwarová PWM dostupná na všech pinech
 - Hardwarová PWM dostupná na GPIO12, GPIO13, GPIO18, GPIO19
- SPI
 - SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
 - SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)
- I2C

- Data: (GPIO2); hodiny (GPIO3)
- EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)
- Sériové
 - TX (GPIO14); RX (GPIO15)



Obrázek 3.1: Raspberry 3 V1.2 (17)

3.1.2 Piano HAT

Piano HAT je vývojový modul ve formátu HAT, který vydala firma Pimoroni v roce 2015. Má 16 kapacitních dotykových tlačítek, z nichž 13 jsou klavírní klávesy, 2 posouvají oktávu nahoru nebo dolů a poslední mění typ nástroje (18). Ve spodní části desky se nachází dvě jména. Prvním je Zachary Igielman, který ve svých 14 letech navrhl Pi Piano, na němž je Piano HAT založeno (19), a druhé patří Philu Howardovi z Pimoroni, který modul navrhl v roce 2015. S Raspberry Pi komunikuje modul pomocí I2C protokolu.



Obrázek 3.2: Piano HAT

3.1.3 Adafruit PiTFT - 320x240 2.8 TFT+Touchscreen

Tento malý displej s velikostí úhlopříčky 2.8 palců a rozlišením 320x240 pixelů s 16-bitovými barvami využívá k detekci dotyku rezistivní vrstvu. Jako modul k Raspberry Pi ho vydala v roce 2014 společnost Adafruit.

Displej využívá SPI rozhraní pro přenos obrazu a další komunikaci, jako je informace o poloze bodu dotyku či o stisknutí tlačítka s Raspberry Pi. Také umožňuje několik módů zobrazení a využívá hardwarové GPIO piny. Díky místu na desce jsou k dispozici čtyři tenké taktické spínače připojené k čtyřem GPIO, které lze použít pro základní uživatelské rozhraní. Nakonec jsem je však nevyužil.



Obrázek 3.3: Adafruit PiTFT

3.2 Sestavení komponent

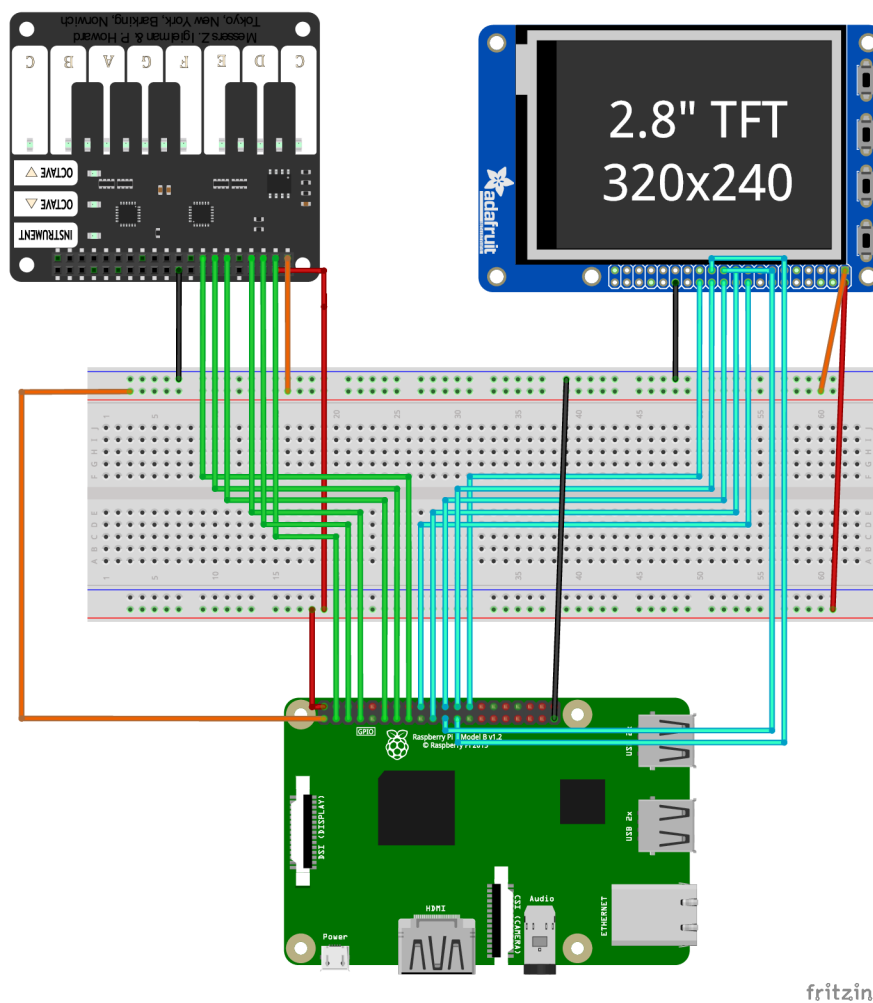
Pro sestavení komponent jsem čerpal z projektu pinout.xyz, který poskytuje databázi GPIO interfaců pro různé HAT moduly a další rozšiřující desky určené pro Raspberry Pi. Výsledné zapojení jsem znázornil pomocí nástroje Fritzing, který umožňuje vytvořit schémata zapojení z již připravených komponent. To však neplatí pro Piano HAT, pro který jsem si musel komponentu připravit sám. (Vektorová vizualizace tohoto modulu je však k dispozici pod vhodnou licencí, takže jsem ji pouze upravil (20)).

3.3 Návrh a vytisknutí pouzdra

Pro zlepšení manipulace s konstrukcí stanice jsem se rozhodl vytvořit vlastní pouzdro, které usnadní manipulaci a interakci s celou stanicí. Inspirací mi byl existující návrh (21), který jsem však upravil tak, aby umožňoval umístění modulů na vnější část pláště.

Pro začátek jsem musel přesně změřit rozměry HAT modulů. S pomocí CAD nástroje Onshape jsem provedl úpravy designu pouzdra. Ty zahrnovaly upravení šířky displeje a výšky dvou HAT modulů pod sebou. Dále jsem do designu integroval otvory pro 2x20pinové female headery a otvory pro uchycení šroubů.

Pro další zlepšení funkcionality a ergonomičnosti pouzdra jsem provedl několik dalších úprav. Nejprve jsem zvážil umístění portů a přístup k nim. Rozhodl jsem se umístit je na levou a boční stranu pouzdra.



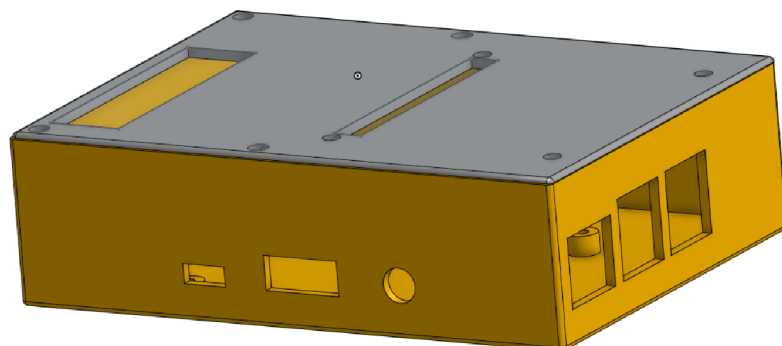
Obrázek 3.4: Sestavení komponent - vizualizace. (poz. Nepájivé pole je zde pouze pro zjednodušení vizualizace. V zařízení není.)

Samotný výrobní proces byl realizován pomocí 3D tiskárny Prusa i3 MK3 a PLA filamentu. Tisk trval zhruba 5 hodin.

Během první iterace pouzdra jsem po vytisknutí objevil několik designových nedostatků, jako například nesprávné umístění některých otvorů na vrchním dílu pouzdra. Tyto nedostatky jsem pak upravil pro další tisky, aby bylo dosaženo optimálního výsledku.

Dále jsem se zaměřil na posílení konstrukce a zvýšení stability pouzdra. Optimalizoval jsem tvar tak, aby bylo dosaženo maximální pevnosti při zachování co nejnižší hmotnosti, a rozšířil mechanismus, který spojuje obě části pouzdra.

Po dalším testování a zhodnocení funkcionality jsem provedl ještě drobné úpravy, jako je upřesnění umístění otvorů pro šrouby a zajištění, že všechny komponenty pouzdra správně sedí a zapadají do sebe.



Obrázek 3.5: Design pouzdra v programu Onshape



Obrázek 3.6: Výsledná podoba sestavení

4 Návrh a implementace aplikace pro stanici

V následující kapitole popisuji instalaci operačního systému na Raspberry Pi a dále návrh a řešení aplikace pro stanici.

4.1 Operační systém a ovladače

Pro Raspberry Pi existuje několik operačních systémů. V zájmu kompatibility a stability OS jsem zvolil Raspbian 10 s kódovým označením Buster. Nastavil jsem uživatelský účet a připojil jsem se k síti. Po úspěšné instalaci jsem provedl aktualizaci systému, abych zajistil, že mám nejnovější verzi všech balíčků. Dále jsem nainstaloval programovací jazyk Python ve verzi 3.9.0 pro vývoj své aplikace.

Následně jsem nainstaloval ovladače pro modul displeje, a to pomocí scriptu `adafruit-pitft.py` (22). Skript napsaný v jazyce Python Melissou LeBlanc-Williams ze společnosti Adafruit Industries je určen k instalaci a konfiguraci displejů Adafruit PiTFT. Je interaktivní a umožňuje zvolit různé možnosti konfigurace, například typ displeje, rotaci, typ instalace (zrcadlení HDMI výstupu a konzole). Zrcadlení z výstupu využívá program `Framebuffer Copy (fbcp)`. Zachycuje data framebufferu (pixels) a kopíruje je na displej PiTFT. Nastavil jsem rotaci displaye na 180°, kvůli designu pouzdra.

Nakonec jsem nainstaloval ovladač Piano HAT (23), který zároveň vyžaduje knihovnu pro kapacitní kontrolery řady `Cap1xxx` od Microchip Technology.

4.2 Funkční požadavky

Pro vypracování této části aplikace jsem si stanovil funkční požadavky, které vycházejí ze zadání:

FP1 - Volná hra: Tato funkce umožní uživatelům hrát hudbu nebo experimentovat s různými zvuky pomocí klávesnice. Uživatelé budou moci změnit používaný hudební nástroj a upravit hlasitost podle svých preferencí.

FP2 - Přihlášení uživatele: Tato funkce umožní uživatelům přihlásit se do systému pomocí účtu, který byl vytvořen učitelem. Přihlášením uživatelé získají přístup ke svým osobním profilům a statistikám.

FP3 - Zobrazení hlavního menu: Po přihlášení se uživatelé dostanou do hlavního menu, které jim poskytne přístup k různým funkcím aplikace, včetně volné hry a seznamu lekcí. Také umožní uživatelům podívat se na jejich dosažené statistiky.

FP4 - Zobrazení všech lekcí: Tato funkce uživatelům umožní procházet seznam všech dostupných lekcí. Uživatelé budou moci vybírat mezi veřejnými lekcemi, které jsou dostupné pro všechny, a soukromými lekcemi, které jsou určeny pouze pro konkrétní skupiny.

FP5 - Zobrazení lekce: Po výběru konkrétní lekce si uživatelé zobrazí obsah této lekce, což je přeměruje na řadu otázek zadaných učitelem.

FP6 - Zobrazení otázky a možnost odpovědi: Během procházení lekcí budou uživatelé dostávat otázky k ověření svých znalostí a porozumění. Tato funkce jim umožní odpovědět na otázky a získat zpětnou vazbu o svém pokroku a úspěšnosti v učení.

FP7 - Aktualizace dat z API: Tato funkce umožní aplikaci stahovat aktuální data z externího rozhraní API. Toto může být užitečné například pro získávání nových lekcí.

FP8 - Offline funkcionality: Tato funkce umožní aplikaci pracovat i v režimu bez připojení k internetu. Když uživatel nebude připojen k síti, aplikace bude stále schopna poskytovat určité funkce a obsah. To zahrnuje možnost hrát hudbu v režimu volné hry, procházet již stažené lekce a provádět činnosti, které nevyžadují online přístup.

4.3 Návrh aplikace

Prvním rozhodnutím bylo zvolení vhodných technologií. Zvolil jsem programovací jazyk Python, a to pro jeho univerzální využití a množství knihoven. Rozhodujícími faktory také byly skutečnosti, že pro již zmíněný modul Piano HAT je napsána knihovna právě v tomto jazyce a moje zkušenost s tímto jazykem. Vzhledem k požadavku na funkci bez připojení k internetu jsem navrhl samostatnou aplikaci, která komunikuje s webovou aplikací pomocí API. Detailnější pohled popisují v diagramu 4.1.

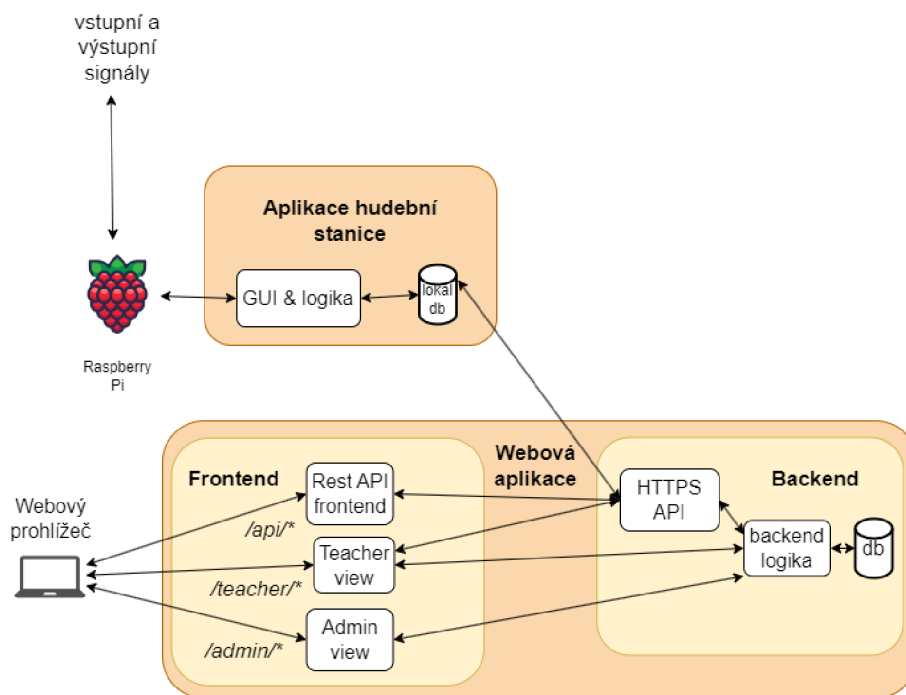
4.3.1 Výběr knihoven a programů pro vytvoření aplikace

Ke grafickému rozhraní jsem se rozhodl využít Pygame. Pygame poskytuje knihovnu pro tvorbu 2D her a interaktivních aplikací v Pythonu. Jeho syntaxe a rozhraní umožňují snadné vytváření uživatelských rozhraní s animacemi, zvuky a dalšími interaktivními prvky, což je ideální pro vytvoření intuitivního rozhraní pro uživatele.

Dále jsem využil knihovnu Mingus pro práci s hudbou. Tato knihovna poskytuje nástroje pro práci s hudebními strukturami, jako jsou akordy, melodie, rytmus a další. Je vhodná pro potřeby mé aplikace, protože umožňuje vytvářet hudební prvky a manipulovat s nimi. Knihovna Mingus také umožňuje integraci s programem FluidSynth, což je softwarový syntezátor MIDI zvuku. Tato integrace poskytuje aplikaci možnost generovat a reprodukovat širokou škálu zvuků a hudebních nástrojů.

4.3.2 Databáze

Pro ukládání a manipulaci s daty v aplikaci jsem se rozhodl využít databázi SQLite. Existuje několik důvodů, proč jsem se rozhodl pro tuto volbu:



Obrázek 4.1: Diagram webové a lokální aplikace

1. **Jednoduchost použití:** SQLite je relační databáze, která nepotřebuje plnohodnotný databázový server. Pro operace s databází je využíváno standardizovaného strukturovaného dotazovacího jazyka SQL, nejrozšířenější volbou pro práci s databází.
2. **Integrace s Pythonem:** Databáze SQLite je integrována do jazyka Python a poskytuje snadný a přímý přístup k databázi pomocí knihovny *sqlite3*. To znamená, že není potřeba instalovat žádné dodatečné knihovny.
3. **Souborová databáze:** SQLite je souborová databáze, což znamená, že je celá uložena v jediném souboru na disku. Toto poskytuje jednoduchost v distribuci naší aplikace, protože není potřeba řešit složité nastavení databázového serveru nebo konfigurace připojení.

4.4 Implementace aplikace

Vstupním bodem je soubor `main.py`, který obsahuje hlavní smyčku programu a základní funkce pro vykreslování a obsluhu událostí. Využívá se knihovna `pygame` pro vykreslování grafického rozhraní a zpracování událostí, jako je kliknutí myši nebo stisk klávesy. Uvnitř smyčky se nachází stavový automat, který určuje, která část aplikace se má vykreslit a obsloužit, a reference na objekty tříd `Button`, `Textbox` a `PopUpMessageBox`, které se mají vykreslit.

V jednotlivých funkcích jsou volány funkce z ostatních souborů, které obsahují třídy a funkce pro jednotlivé části aplikace. V souboru `key_input.py` se nachází třída `KeyInput`, která slouží k zpracování stisknutých kláves a převodu na noty.

Soubor `db.py` obsahuje funkce pro práci s databází, včetně vytvoření tabulek, vkládání, mazání a získávání dat. Obsahuje také funkce pro přihlášení a odhlášení studenta a práci s REST API včetně získávání synchronizace dat s webovou aplikací.

Soubor `virtual_keyboard.py` obsahuje třídu `VirtualKeyboard`, která slouží k vykreslení virtuální klávesnice. Ta se zobrazuje při zadávání textu.

4.4.1 Implementace vlastních objektů grafického rozhraní

Pro interakci uživatele s aplikací jsem vytvořil vlastní objekty grafického rozhraní, jako jsou tlačítka, textová pole a vyskakovací okna. Tyto objekty jsou reprezentovány třídami `Button`, `Textbox` a `PopUpMessagebox`, které mohou být dále rozšířeny o další funkce a vlastnosti. Například jsem vytvořil třídu `SwitchButton`, která umožňuje přepínání mezi dvěma stavy, a třídu `FramedButton`, která slouží k vykreslení tlačítka s rámečkem. Třída `Button` obsahuje metody pro vykreslení tlačítka a zpracování kliknutí myši. Třída `Textbox` slouží k vykreslení textového pole a zpracování vstupu od uživatele. Třída `PopUpMessagebox` slouží k vykreslení vyskakovacího okna s textem a tlačítky pro potvrzení nebo zrušení akce.

4.4.2 Dotyková klávesnice a klaviatura

Pro zpracování vstupu z klávesnice jsem vytvořil třídu `KeyInput`, která obsahuje metody pro zpracování stisknutých kláves a převod na noty. Tato třída je využívána v rámci aplikace pro hru na klavír a zadávání textu do textových polí. Knihovna `pianohat` využívá „`handlers`“, což jsou lambda funkce, které jsou volány při stisknutí klávesy a slouží k zpracování vstupu. Tímto způsobem lze snadno rozšířit funkcionalitu aplikace o další možnosti interakce s uživatelem. Dotyková klávesnice je vykreslena pomocí třídy `VirtualKeyboard`, která obsahuje metody pro vykreslení klávesnice a zpracování kliknutí myši. Tato klávesnice se zobrazuje při zadávání textu do textových polí a umožňuje uživateli snadněji zadávat text pomocí myši nebo dotykové obrazovky.

4.4.3 API a databáze

Pro práci s databází jsem vytvořil modul `db.py`, který obsahuje funkce pro vytvoření tabulek, vkládání, mazání a získávání dat z databáze. Databáze obsahuje tabulky pro uživatele, lekce, otázky a odpovědi. Pro získávání dat z databáze jsem použil SQL dotazy a knihovnu `sqlite3`. K ukládání offline dat využívám tabulku `student_lesson_offline`, která obsahuje data o lekcích, které byly dokončeny offline a čekají na synchronizaci s webovou aplikací. Pro synchronizaci dat s webovou aplikací jsem vytvořil funkce pro získávání a odesílání dat pomocí REST API a knihovny `requests`. K přihlášení využívám funkci `login`, která ověřuje uživatele pomocí jména a hesla a ukládá ID uživatele do databáze. Po dotazu na API získá aplikace token, který je uložen do databáze a používán pro další dotazy na API. Pro odhlášení uživatele slouží funkce `logout`, která odstraní token z databáze a odhlásí uživatele.

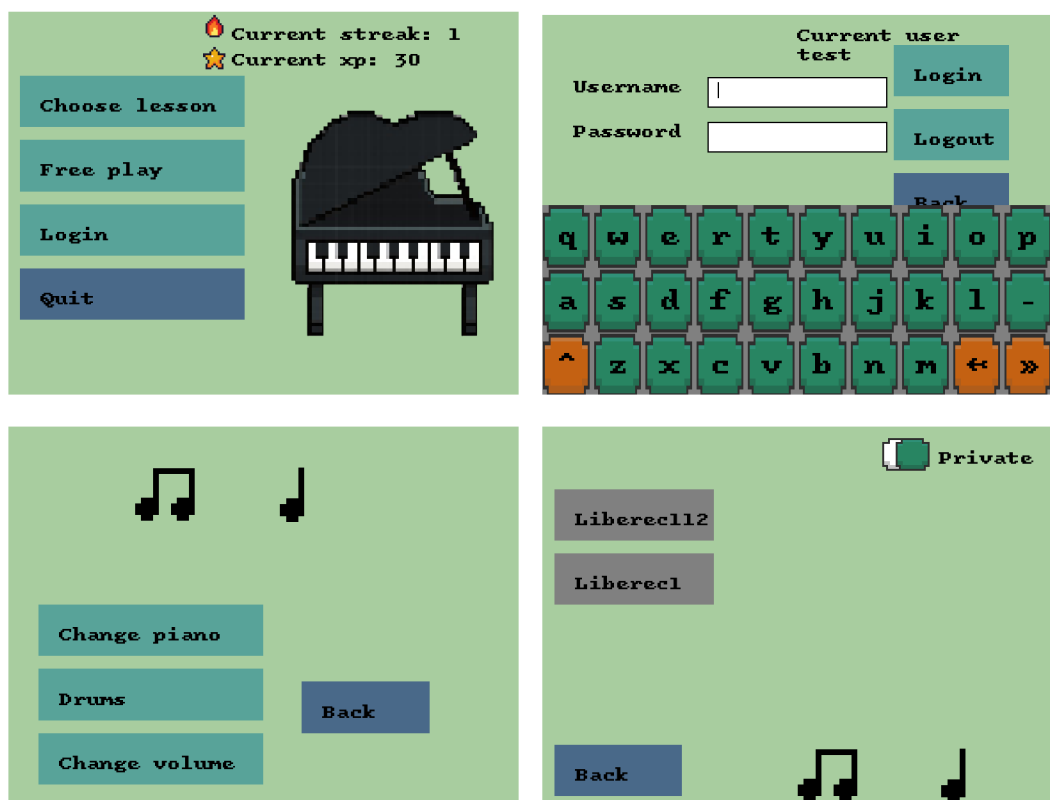


Obrázek 4.2: Databázový diagram

4.4.4 Grafické rozhraní

GUI jsem se snažil přispůsobit malým dotykovým obrazovkám, proto jsem vytvořil tlačítka s velkým písmem a dostatečným odstupem mezi nimi. Rozlišení obrazovky je 800x600 pixelů, což je dostatečné pro zobrazení všech potřebných aplikací. Ačkoliv má displej menší rozlišení, operační systém umožňuje zmenšit obrazovku z vyššího rozlišení. Poměr a rozlišení jsou pevně nastaveny a nemění se, vzhledem k tomu, že aplikace je určena pro určitý typ zařízení s konkrétním displejem. Grafické rozhraní zahrnuje:

1. Hlavní menu, které obsahuje tlačítka pro výběr lekce, volnou hru, přihlášení a odhlášení uživatele a ukončení aplikace. Na obrazovce je zobrazena aktuální série (tak-



Obrázek 4.3: Snímky grafického rozhraní

zvaný streak) a počet získaných bodů zkušenosti (takzvané xp).

2. Menu, kde jsou zobrazeny lekce, které si uživatel může vybrat. Tlačítka jsou zbarvena podle toho, zda již byla dokončena. Uživatel si může vybrat lekci a začít ji plnit. Po jejím dokončení se zobrazí vyskakovací okno s informacemi o získaných bodech a možnosti návratu do hlavního menu.
3. Hra na klavír, kde může uživatel volně improvizovat. Má možnost měnit zvuk pi-ana, hlasitost nebo přehrát bicí smyčku. Po stisknutí klávesy se zobrazí noty na obrazovce a zazní zvuk noty.
4. Přihlašovací obrazovka, kde uživatel může zadat jméno a heslo a přihlásit se do aplikace. Využívá se virtuální klávesnice pro zadávání textu do polí pro přihlašovací údaje. Po přihlášení se zobrazí vyskakovací okno s informací o úspěšném přihlášení nebo chybě.

4.4.5 Generování notového zápisu pomocí Lilypond

Pro generování notového zápisu jsem využil knihovnu Lilypond, která umožňuje vytvářet notové zápisy pomocí textového formátu. Pro vytvoření otázky se vygeneruje notový zápis na základě správné odpovědi a následně se zobrazí na obrazovce. Pro vybrání pouze

náhledu notového zápisu jsem využil parametr „-dcrop“ při generování obrázku. Knihovna Mingus umožňuje převést objekt Bar na Lilypond textový řetězec, který je následně předán aplikaci Lilypond pro vykreslení notového zápisu. Tímto způsobem lze generovat notové zápisy pro různé typy otázek, jako je identifikace not nebo akordů.

4.4.6 Syntetizace zvuku pomocí FluidSynth

Pro syntetizaci zvuku jsem využil integraci knihovny Mingus s aplikací FluidSynth. Tímto způsobem lze generovat zvuky not a akordů na základě uživatelského vstupu. Využil jsem také metodu `set_instrument`, která umožňuje změnit zvukový nástroj pomocí MIDI programu a banky zvuků. Pro správnou funkci syntetizace zvuku je nutné mít nainstalovaný FluidSynth a banku zvuků, která obsahuje zvuky jednotlivých nástrojů. K instalaci aplikace doporučuji použít balíčkovací systém apt a nainstalovat balíčky `fluidsynth` a `fluid-soundfont-gm`. Experimentálně jsem také zjistil, že je potřeba mít nainstalovaný balíček `jackd1`, rovněž dostupný v repozitářích raspbianu.

5 Návrh a řešení webové aplikace

V následující kapitole jsem popsal návrh a řešení webové aplikace, která umožňuje učitelům vytvářet a spravovat lekce a otázky pro své studenty. Aplikace také umožňuje studentům získat informace o svých lekcích a odpovědět na otázky. Také obsahuje API pro získání data ve formátu JSON a jejich zobrazení na webu s grafickým rozhraním.

5.1 Funkční požadavky

Pro návrh webové aplikace bylo nutné nejprve stanovit funkční požadavky, které by měla aplikace splňovat. Tyto požadavky jsem zvolil na základě zadání a vlastního uvážení. Funkční požadavky jsou následující:

5.1.1 Modul učitele

WFP1 - Operace s uživatelským účtem: Tato funkce umožní učitelům a administrátorům zaregistrovat a přihlásit se, měnit své heslo, získat zapomenuté heslo, upravit další profilové informace a odhlásit se.

WFP2 - Operace s uživateli: Tato funkce umožní učitelům získat seznam svých studentů, zobrazit informace o studentech a vytvořit nové studenty.

WFP3 - Operace s lekcemi: Tato funkce umožní učitelům získat seznam lekcí, zobrazit detaily lekce a získat seznam studentů, kteří dokončili lekci.

WFP4 - Operace s otázkami: Tato funkce umožní učitelům získat seznam otázek a zobrazit detaily otázky.

WFP5 - Dashboard: Tato funkce umožní učitelům zobrazit přehled o svých studentech, včetně informací o tom, kolik času strávili na lekcích, kolik mají XP bodů a kolik lekcí dokončili.

5.1.2 Modul API

WFP6 - Zobrazení API ve formátu JSON: Tato funkce umožní uživatelům zobrazit data ve formátu JSON.

WFP7 - Zobrazení API na webové stránce: Tato funkce umožní uživatelům zobrazit data na webové stránce s grafickým rozhraním.

WFP8 - Získání lekcí: Tato funkce umožní uživatelům získat seznam lekcí, které mohou

splnit.

WFP9 - Získání informací o lekci: Tato funkce umožní uživatelům získat informace o konkrétní lekci včetně otázek, které mají zodpovědět.

WFP10 - Aktualizace informací o lekci: Tato funkce umožní uživatelům aktualizovat informace o lekci, jako je čas strávený na lekci, počet zodpovězených otázek a počet správných odpovědí.

WFP11 - Přihlášení pomocí API: Tato funkce umožní uživatelům přihlásit se pomocí API a získat přístupový token.

5.1.3 Modul administrátora

WFP12 - Správa všech objektů: Tato funkce umožní administrátorům spravovat všechny objekty v aplikaci, jako jsou uživatelé, lekce, otázky a studenti.

5.2 Ostatní požadavky

Pro návrh webové aplikace bylo nutné nejprve stanovit nefunkční požadavky, které by měla aplikace splňovat. Tyto požadavky jsem zvolil na základě zadání a vlastního uvážení. Nefunkční požadavky jsou následující:

WOP1 - Bezpečnost: Aplikace musí být zabezpečena proti neoprávněnému přístupu a útokům.

WOP2 - Rychlost: Aplikace musí být rychlá a efektivní, aby uživatelé mohli rychle získat požadované informace.

WOP3 - Uživatelská přívětivost: Aplikace musí být snadno použitelná a přehledná, aby uživatelé mohli snadno najít požadované informace.

5.3 Návrh aplikace

Rozhodnutí o technologiích jsem učinil na základě svých znalostí a zkušeností s danými technologiemi. Jelikož jsem zvolil programovací jazyk Python při návrhu aplikace pro hudební stanice, rozhodl jsem se použít stejný jazyk pro návrh webové aplikace. Pro vývoj webové aplikace jsem zvolil velmi populární framework Django. Ten nabízí například funkce pro autentizaci uživatelů, správu databáze a templatovací systém webových stránek. Zároveň Django obsahuje ochranu proti běžným útokům, jako je SQL injection a cross-site scripting (24) a měl jsem s ním již zkušenosti z předchozích projektů.

5.4 Výběr knihoven

K naprogramování API modulu jsem zvolil knihovnu Django REST framework, která poskytuje snadný způsob, jak vytvářet API ve formátu JSON. Crispy forms jsou další knihov-

nou, kterou jsem zvolil pro tvorbu formulářů v aplikaci. Tato knihovna poskytuje snadný způsob, jak vytvářet formuláře s pomocí Bootstrapu (Bootstrap je CSS framework) a je užitečná pro tvorbu uživatelského rozhraní. Zároveň jsem zvolil knihovnu Font Awesome, která poskytuje ikony pro webové stránky a je vhodná pro zlepšení uživatelského rozhraní.

5.5 Výběr databáze

Pro ukládání dat jsem zvolil databázi SQLite, kterou jsem popsal v podkapitole 4.3.2. V případě, že by bylo potřeba rozšířit aplikaci a zvýšit výkon, bylo by možné přepnout na databázi PostgreSQL nebo MySQL. Pro vytvoření databázového modelu jsem využil Django ORM, který umožňuje definovat databázové modely pomocí Python tříd a metod (24).

5.6 Implementace aplikace

Aplikaci jsem rozložil do modulů `lessons_api` a `teachers_view`. Modul `lessons_api` obsahuje modely pro lekce a otázky, modul `teachers_view` obsahuje webovou aplikaci určenou pro učitele. Každý modul obsahuje soubor `urls.py`, kde jsou definovány URL cesty. V souboru `settings.py` jsem nastavil databázi, rozšíření, logování a další konfigurace aplikace.

5.6.1 Django konvence

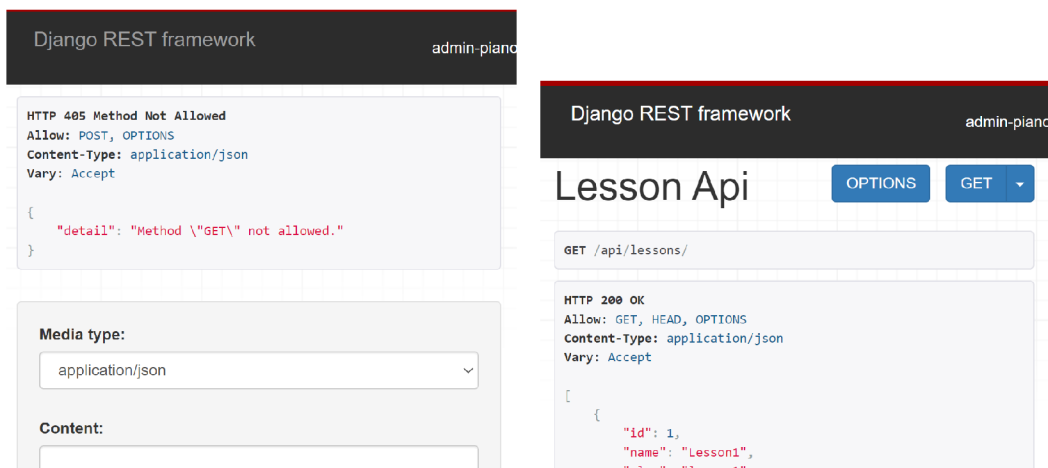
Django používá koncept projektu a aplikací (Já pro výraz aplikace používám v dalších kapitolách textu pojem moduly), kvůli lepší čitelnosti kódu. Projekt obsahuje jednu nebo více aplikací, které v rámci něj vykonávají různou funkci.

Nejdříve jsem vytvořil projekt pomocí příkazu `django-admin startproject <název projektu>`. Každá aplikace, která je vytvořena pomocí příkazu `python manage.py startapp <název aplikace>` a obsahuje soubory při vygenerování: administrátorské rozhraní - `admin.py`, aplikace - `apps.py`, modely - `models.py` a pohledy - `views.py`, je přidána do `settings.py` v sekci `INSTALLED_APPS`. Další soubory, které jsou často využívány, jsou `urls.py`, kde jsou definovány URL cesty a `serializers.py`, kde jsou definovány serializéry pro modely. Také složka `templates`, která obsahuje HTML šablony a `static`, která obsahuje CSS, JS a obrázky (24). Pro Crispy forms jsem vytvořil soubor `forms.py`, kde jsem definoval formuláře pro aplikaci.

5.6.2 Modul API

Modul API využívá Django REST framework pro vytvoření API ve formátu JSON. Zprv je využit v souboru `serializers.py`, kde jsem definoval serializéry pro modely lekcí a otázek. V souboru `views.py` jsem vytvořil pohledy pro zobrazení dat a operace s daty.

Zde využívám dekorátor `@api_view`, který umožňuje definovat, jaký typ HTTP requestu je povolen a dekorátory `@authentication_classes` a `@permission_classes`, které definují autentizaci pro přístup k datům (25). Zadruhé jsem v `urls.py` definoval URL cesty, které definuji v tabulce 5.1.



Obrázek 5.1: Snímky grafického rozhraní modulu API

URL cesta	Funkce
api/lessons/	Získání seznamu lekcí
api/lessons/<pk>/	Získání detailů lekce
api/questions/	Získání seznamu otázek
api/students/time-spent/	Získání času stráveného studenty
api/students/xp/	Získání XP bodů studentů
api/students/lessons-finished/	Získání počtu dokončených lekcí studentů
api/self/	Získání informací o uživateli
api/students/	Získání seznamu studentů

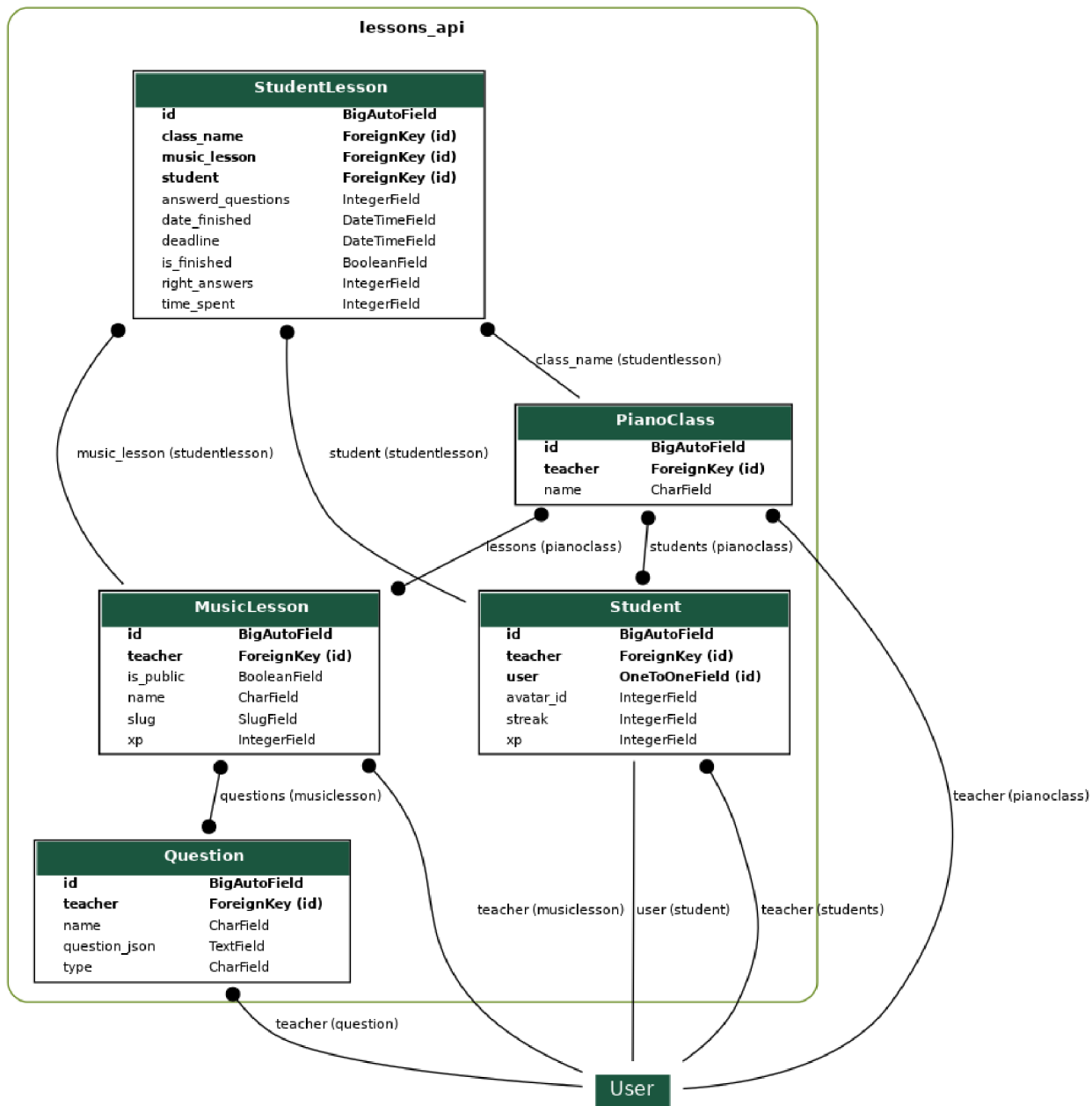
Tabulka 5.1: URL cesty API

Zatřetí jsem v `models.py` definoval modely pro lekce, otázky, studenty a hudební třídy. Zároveň jsem zde naprogramoval signál pro vytvoření autentizačního tokenu pro nového uživatele. Ve výchozím nastavení vrátí rozhraní API formát určený hlavičkami. V případě prohlížeče je to HTML. Formát lze zadat pomocí `?format=` v požadavku. Takže pro odpověď ve formátu JSON lze v prohlížeči získat přidáním `?format=json` do adresy URL (25).

5.6.3 Modul učitele

V modulu učitele jsem vytvořil uživatelské rozhraní včetně formulářů pro vytváření lekcí a otázek, zobrazení informací o studentech a lekcích a dashboard pro zobrazení přehledu o studentech. V souboru `views.py` jsem vytvořil pohledy pro operace s uživatelským účtem, uživateli, lekcemi, otázkami a dashboardem.

Zde jsem využil dekorátory `@login_required`, které umožňují definovat pohledy, které vyžadují přihlášení uživatele. V `urls.py` jsem definoval URL cesty, které defínuji v tabulce 5.2.



Obrázek 5.2: Databázové schéma webové aplikace

URL cesta	Funkce
/	Dashboard
teacher/dashboard/	Dashboard
teacher/lessons/	Seznam lekcí
teacher/lessons/<pk>/edit/	Editace lekce
teacher/lessons/<pk>/delete/	Smazání lekce
teacher/lessons/create/	Vytvoření lekce
teacher/students/	Seznam studentů
teacher/students/create/	Vytvoření studenta
teacher/settings/	Nastavení
teacher/settings/delete/	Smazání účtu
teacher/question/create/	Vytvoření otázky
teacher/login/	Přihlášení

Tabulka 5.2: URL cesty pro modul učitele

5.6.4 Grafické rozhraní modulu učitele

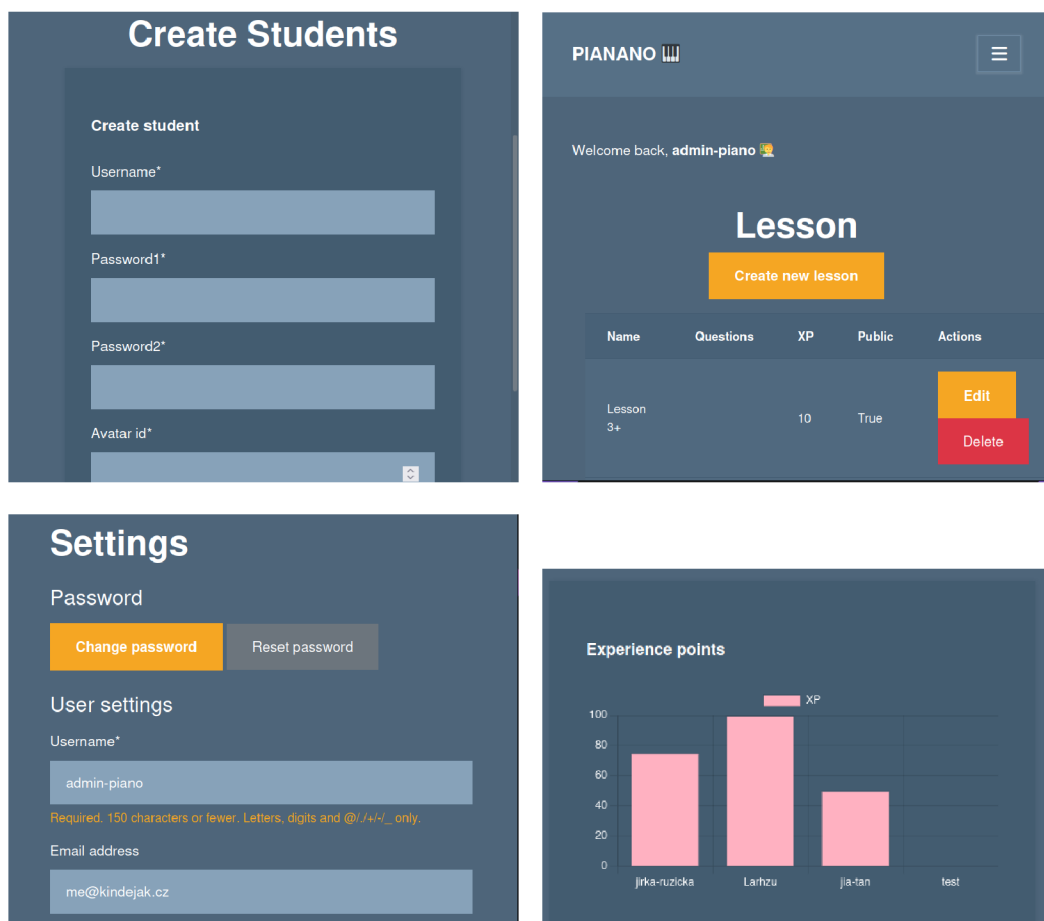
V grafickém rozhraní modulu učitele jsem se zaměřil na uživatelskou přívětivost a snadnou navigaci mezi jednotlivými funkcemi. K vykreslování používám templatovací systém frameworku Django. Template obsahuje proměnné, které se při vyhodnocení šablony nahradí hodnotami, a značky, které řídí logiku šablony. Dále používám javascriptovou knihovnu chartjs, pro vykreslování grafů.

Navigace probíhá pomocí menu v horní části webu. Pokud není uživatel přihlášen, je přesměrován na stránku pro přihlášení a po úspěšném přihlášení má přístup k následujícím může přistoupit k následujícím možnostem:

- **Dashboard:** Na úvodní stránce učitelského rozhraní se nachází dashboard poskytující přehledné informace o aktivitě studentů. V grafech jsou zde zobrazeny statistiky o zkušenostních bodech a aktuální streak učení.
- **Seznam lekcí:** Umožňuje učiteli procházet všechny vytvořené lekce. Z této sekce je také možné přistoupit k editaci nebo smazání konkrétní lekce a vytvoření nové lekce.
- **Seznam studentů:** Poskytuje přehled všech studentů zapojených do kurzu. Každý student je zde identifikován jménem či přezdívkou a případně dalšími informacemi jako je například pokrok v kurzu. Zároveň je zde možné přidat studenta do kurzu.
- **Nastavení:** Tato sekce umožňuje učiteli upravit svůj uživatelský účet. Lze změnit heslo nebo upravit osobní údaje, popřípadě celý účet smazat.

5.7 Nasazení a zabezpečení

Pro zabezpečení přenosu dat mezi klienty a serverem využívá aplikace protokol HTTPS. Tento protokol zajišťuje šifrovanou komunikaci pomocí certifikátů, což minimalizuje



Obrázek 5.3: Snímky grafického rozhraní webové aplikace

riziko útoků typu MITM a ochraňuje před odposlechem dat. Implementace HTTPS je realizována pomocí Nginx proxy manageru, který funguje jako reverzní proxy server (26). Tímto způsobem je veškerý provoz směrován přes HTTPS, což zvyšuje bezpečnost komunikace.

Certifikáty používané pro šifrování komunikace jsou vydávány certifikační autoritou Let's Encrypt a jsou obnovovány automaticky jednou za tři měsíce. Tento postup zajišťuje platnost certifikátů a minimalizuje ruční zásahy do jejich správy, což snižuje riziko zapomenutí na obnovu certifikátů a narušení provozu aplikace. Pro zabezpečení API endpointů a celkové autentizace uživatelů je využíván systém zabudovaný do Django frameworku. Autentizace je prováděna pomocí kombinace uživatelských jmen a hesel a tokenu v případě API. Pro zajištění bezpečnosti dat je implementován pravidelný proces zálohování dat a možnost obnovy v případě potřeby. Zálohování je prováděno automaticky prostřednictvím cloudového zprostředkovatele Oracle.

5.7.1 Nasazení

Pro účely nasazení aplikace do produkčního prostředí jsem implementoval automatizovaný proces pomocí vzdáleného SSH přístupu. Tento proces probíhá pomocí platformy Gi-

tHub pro správu verzovacího nástroje Git. Github Actions a plugin appleboy/ssh-action, umožňují spuštění příkazů na vzdáleném serveru po provedení push operace do repozitáře. Proces nasazení aplikace se skládá z následujících kroků:

1. **Použití GitHub Actions triggeru:** Nasazení je spuštěno automaticky po detekci commitu obsahujícího specifickou zprávu („please deploy“).
2. **Připojení k vzdálenému serveru pomocí SSH:** Akce se připojí k serveru pomocí SSH klíčů a přihlašovacích údajů uložených v tajných proměnných (anglicky secrets), ke kterým má přístup pouze vlastník repozitáře.
3. **Aktualizace kódu z repozitáře:** Proveďte se aktualizace kódu aplikace z hlavní větve repozitáře pomocí příkazu git pull.
4. **Aktualizace konfigurace:** Nastaví se konfigurační soubor .env s nezbytnými proměnnými prostředí.
5. **Spuštění kontejnerů pomocí Docker Compose:** Aktualizovaná aplikace je nasazena pomocí Docker Compose, což zahrnuje vytvoření obrazů aplikace a spuštění kontejnerů definovaných v docker-compose.yml.

6 Testování a návrhy na zlepšení

V následující kapitole jsem popsal proces testování výsledného produktu, tedy vzdělávací stanice pro hudební teorii a přidružené webové aplikace. Testování hrálo klíčovou roli při zajištění kvality a spolehlivosti produktu a probíhalo průběžně během celého vývoje. Jedním z hlavních cílů testování bylo identifikovat chyby a nedostatky v funkcionalitě, uživatelském rozhraní a výkonu aplikace a následně je opravit.

6.1 Testování

Manuální testování bylo klíčovou metodou, kterou jsem použil k důkladnému ověření funkcionality stanice a webové aplikace. Testování zahrnovalo ruční otestování jednotlivých funkcí, interakcí s uživatelským rozhraním a ověřování správnosti výstupů.

Použití debugovacích zpráv mi umožnilo rychle identifikovat chyby v kódu a zjistit příčiny nežádoucího chování aplikace. Vložением zpráv do kódu v klíčových místech jsem sledoval, jak data proudí skrz aplikaci ve snaze zjistit, kde se vyskytují problémy.

Během testování byly identifikovány různé chyby a nedostatky. Mezi nejčastěji se vyskytující problémy patřily:

- **Chyby v uživatelském rozhraní:** Často se vyskytovalo nesprávné zobrazení prvků, chybějící popisky nebo nekonzistentní chování.
- **Chyby ve funkcionalitě:** Některé funkce stanice a webové aplikace nepracovaly správně nebo nedodržovaly funkční požadavky.

Všechny nedostatky byly následně opraveny v průběhu vývoje.

6.2 Testování z pohledu uživatele

Během testování z pohledu uživatele byly vytvořeny různé uživatelské scénáře, které simulovaly běžné situace. Požádal jsem mnou vybrané testery, aby prováděli různé úkoly a v průběhu testování jsem sbíral jejich reakce. V souhrnu jsem zjistil, že:

- Celkové reakce byly pozitivní, testerům se líbila grafická stylizace, prvky gamifikace a syntetizace zvuku.
- Webová aplikace je podle testerů přehledná.

- Klaviatura má rozsah pouze jednu oktávu, což značně omezuje možnosti hraní skladeb. Zároveň funguje pouze na základě dotyku, takže nelze trénovat například sílu úhozu.
- Dotykový displej je také malý, takže se občas špatně obsluhuje.

6.3 Návrhy na zlepšení

Na základě zjištěných chyb a nedostatků během testování, stejně jako na základě zpětné vazby od uživatelů, byly navrženy možnosti vylepšení.

6.3.1 Rozšíření lekcí a přidání hudební teorie

Testovaná verze obsahovala omezený počet lekcí a cvičení, což by mohlo omezit pokrok studentů a motivaci k učení. Rozšíření obsahu by poskytlo více možností pro rozvoj svých hudebních dovedností. Dalším zlepšením by bylo přidání více hudební teorie do vzdělávacího obsahu.

6.3.2 Další prvky gamifikace

Vzhledem k pozitivním reakcím testerů na prvky gamifikace je mým návrhem přidání dalších herních prvků, které by motivovaly uživatele k pravidelnému učení a zdokonalování svých dovedností. To by mohlo zahrnovat odměny za dosažení určitých cílů, soutěže mezi uživateli nebo možnost sbírání virtuálních odznaků za úspěšné dokončení úkolů.

6.3.3 Využití vlastní desky

Využití vlastní desky by přineslo několik výhod. První z nich je zmenšení prostoru pro hardware, což by vedlo k menšímu prostorovému nárokům a kompaktnějšímu designu vzdělávací stanice. Další výhodou by byl optimalizovaný hardware, což by umožnilo plynulejší provoz aplikace. Vytvořením vlastních výstupních obvodů by umožnilo například připojení vlastních MIDI kláves. Nakonec, přidání akumulátoru by zvýšilo přenosnost produktu, což by zvyšovalo jeho flexibilitu a mobilitu.

7 Závěr

Hlavním cílem této bakalářské práce bylo navrhnout a implementovat zařízení, které by zábavnou a interaktivní formou umožnilo studentům procvičovat si své znalosti a schopnosti hry na piano. Prvním krokem bylo provést rešerši o historii klavíru a metodách jeho výuky, abych porozuměl celkovému kontextu. Poté jsem zkoumal historické i současné trendy v oblasti vzdělávání a využití gamifikace ke zlepšení učebního procesu. Následně jsem navrhl propojení hardwarových komponent a pouzdro, které jsem následně vytiskl na 3D tiskárně. Základním stavebním kamenem je počítač Raspberry Pi, který je propojen s klaviaturou a dotykovým displejem.

V dalším kroku jsem navrhl a implementoval dvě aplikace. Pro aplikaci na Raspberry Pi zařízení jsem využil Pygame, což mi umožnilo vytvořit uživatelsky přívětivé prostředí pro interaktivní učení hry na klavír. Druhou aplikací je webová platforma vyvinutá pomocí frameworku Django. Tato webová aplikace slouží k administraci a správě výuky hry na klavír. Django mi poskytl silný základ pro vytvoření komplexního webového rozhraní, které zahrnuje moduly pro učitele, administrátory a rozhraní API pro komunikaci se stanicí. Během implementace jsem kladl důraz na uživatelský komfort a rozšiřitelnost lekcí.

Po dokončení implementace jsme provedli testování aplikace jak z technického hlediska, tak i z hlediska uživatelského přijetí. Zpětná vazba mi poskytla cenné informace pro další vývoj a zdokonalení aplikace.

Věřím, že toto zařízení představuje nástroj pro podporu výuky hry na klavír, který může motivovat studenty a zlepšit jejich schopnosti a zároveň tento projekt nabízí základ, který může být v budoucnu rozšířen.

Bibliografie

1. HAVRÁNEK, B. *Slovník spisovného jazyka českého*. 2011. vyd. Ústav pro jazyk český, 2011. Dostupné také z: <https://ssjc.ujc.cas.cz/search.php>.
2. SÝKORA, V.J. *Dějiny klavírního umění od nejstarší doby až po současnost*. Panton, 1973. Dostupné také z: <https://ndk.cz/uuid/uuid:8f4cf4a0-bc23-11e4-9541-005056827e51>.
3. KAPP, K.M. *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. Wiley, 2012. Business book summary. ISBN 9781118096345.
4. LEE, J.; HAMMER, J. Gamification in education: What, how, why bother? *Academic exchange quarterly*. 2011, roč. 15, č. 2, s. 146.
5. ARENSON, Michael A.; HOFSTETTER, Fred T. High-Tech Models for Music Learning: The GUIDO System and the PLATO Project. *Music Educators Journal* [online]. 1983, roč. 69, č. 5, s. 46–51 [cit. 2024-05-02]. ISSN 00274321, ISSN 19450087. Dostupné z: <http://www.jstor.org/stable/3396201>.
6. DANNENBERG, Roger B; SANCHEZ, Marta; JOSEPH, Annabel; CAPELL, Peter; JOSEPH, Robert; SAUL, Ronald. An expert system for teaching piano to novices. In: *ICMC*. 1990.
7. DANNENBERG, Roger B; SANCHEZ, Marta; JOSEPH, Annabel; CAPELL, Peter; JOSEPH, Robert; SAUL, Ronald. A computer-based multi-media tutor for beginning piano students. *Interface*. 1990, roč. 19, č. 2-3, s. 155–173. Dostupné z DOI: [10.1080/09298219008570563](https://doi.org/10.1080/09298219008570563).
8. SANCHEZ, Marta; JOSEPH, Annabelle; DANNENBERG, Roger. *SFCI Archive: The Piano Tutor* [<https://archive.org/details/sfci-archive-the-piano-tutor-1990>]. 1990. [citováno 03-05-2024].
9. *Ricci Adams — ricciadams.com* [<https://www.ricciadams.com/>]. [B.r.]. [citováno 03-05-2024].
10. ADAMS, Ricci. *musictheory.net — musictheory.net* [<https://www.musictheory.net>]. [B.r.]. [citováno 03-05-2024].
11. *Yousician - FUTURIUM - European Commission — ec.europa.eu* [<https://ec.europa.eu/futurium/en/innovative-science-2019/yousician.html>]. [B.r.]. [citováno 13-05-2024].

12. YUN, Yi Tan; THIRUVARUL, Sinthu. Understanding the potential of music learning application as a tool for learning and practicing musical skills. *International Journal of Creative Multimedia*. 2021, roč. 2, č. 1, s. 42–56.
13. DUOLINGO. *Our brand-new Music course hits all the right notes* — *blog.duolingo.com* [<https://blog.duolingo.com/music-course/>]. [B.r.]. [citováno 13-05-2024].
14. MONK, S. *Raspberry Pi Cookbook: Software and Hardware Problems and Solutions*. O'Reilly Media, 2016. ISBN 9781491939055. Dostupné také z: <https://books.google.cz/books?id=0skvDAAAQBAJ>.
15. RASPBERRY PI, Ltd. *Raspberry Pi 3 Model B* [<https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>]. [B.r.]. [citováno 04-05-2024].
16. RASPBERRY PI, Ltd. *Raspberry Pi Documentation* [<https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>]. [B.r.]. [citováno 04-05-2024].
17. HALFACREE, Gareth. *File:Raspberry Pi 3 (24914484549).png* — *Wikimedia Commons, the free media repository*. 2022. Dostupné také z: [https://commons.wikimedia.org/w/index.php?title=File:Raspberry_Pi_3_\(24914484549\).png&oldid=643980702](https://commons.wikimedia.org/w/index.php?title=File:Raspberry_Pi_3_(24914484549).png&oldid=643980702). [Online; citováno 4-May-2024].
18. MACDONALD, Sandy. *Piano HAT* [<https://sandyjmacdonald.github.io/2015/08/10/piano-hat/>]. [B.r.]. [citováno 04-05-2024].
19. LYNN, Helen. *PiPiano: a musical, educational add-on board* [[HelenLynn](#)]. [B.r.]. [citováno 04-05-2024].
20. *GitHub - amperka/hardware-drawings* [<https://github.com/amperka/hardware-drawings>]. [B.r.]. [citováno 04-05-2024].
21. PETZOLD, Tom. *Free CAD Designs, Files & 3D Models | The GrabCAD Community Library* — *grabcad.com* [<https://grabcad.com/library/raspberry-pi-3b-simple-case-1>]. [B.r.]. [citováno 05-05-2024].
22. LEBLANC-WILLIAMS, Melissa. *Raspberry-Pi-Installer-Scripts/adafruit-pitft.py at main · adafruit/Raspberry-Pi-Installer-Scripts* — *github.com* [<https://github.com/adafruit/Raspberry-Pi-Installer-Scripts/blob/main/adafruit-pitft.py>]. [B.r.]. [citováno 09-05-2024].
23. PIMORONI, Ltd. *GitHub - pimoroni/Piano-HAT: Python library and examples for Piano HAT Raspberry Pi Add-on board* — *github.com* [<https://github.com/pimoroni/Piano-HAT>]. [B.r.]. [citováno 09-05-2024].
24. FOUNDATION, Django. *Django documentation contents | Django documentation* — *docs.djangoproject.com* [<https://docs.djangoproject.com/en/5.0/contents/>]. [B.r.]. [citováno 14-05-2024].
25. CHRISTIE, Tom. *Home - Django REST framework* — *django-rest-framework.org* [<https://www.django-rest-framework.org/>]. [B.r.]. [citováno 14-05-2024].
26. CURNOW, Jamie. *Nginx Proxy Manager* — *nginxproxymanager.com* [<https://nginxproxymanager.com/guide/>]. [B.r.]. [citováno 14-05-2024].

A Přílohy

Pro vylepšení stylistiky jsem využil asistenci LLM modelu umělé inteligence ChatGPT-3.5.

A.1 Zdrojové kódy

Zdrojový kód je k dispozici ve dvou repozitářích na Githubu:

- <https://github.com/kindejak/Pianano>
- <https://github.com/kindejak/pianano-gui>

A.2 Sestavení komponent - schéma

